DOCUMENT RESUME

ED 031 066 EF 003 543

By-Murphy, Judith; Sutter, Robert

School Scheduling by Computer, the Story of GASP.

Educational Facilities Labs., Inc., New York, N.Y.
Pub Date Dec 64
Note-64p.
EDRS Price MF-$0.50 HC-$3.30
Descriptors-*Computer Programs, *Data Processing, *Flexible Schedules, *Programers, *School Schedules

Generalized academic simulation programs (GASP) is school scheduling by computer. The publication compares the manual method of developing a school schedule with the new method of computer programing. A major advantage of computer programing is that it relieves the administrator of this busy work so that he can devote his energies and time to more important educational problems. To convert to computer scheduling, a school principal will have to become acquainted with computer language or employ a programer in a data process center. A major advantage to computer programing is that it can facilitate flexible scheduling of pupils, heretofore stymied by human shortcoming and inadequacy. Computer scheduling allows considerable leeway in setting the parameters for the scheduling of subjects. GASP should not be viewed as limited cybernetically to the scheduling of pupils. Once an administrator understands its language and its potential, the computer can be used to answer "what if" questions and "educational assumptions" which without the computer the administrator could only imagine but never implement for lack of computer programing and the knowledge of outcomes. This document previously announced as ED 018 092. (JZ)

SCHOOL SCHEDULING BY COMPUTER ■ THE STORY OF GASP

A REPORT FROM EDUCATIONAL FACILITIES LABORATORIES

IS RIDGEWOOD 12

	IX	NSUB	NDIV	SCHN	NSCH	MEAS	LOAD
11020	626	12	24	1	15	2.00	83
80010	1	16	41	10	15	2.00	91
80020	2	14	36	1	1	.00	87
80050	5	16	46	2	2	.00	87
80080	8	14	36	1	1	.00	87
80320	32	16	47	1	1	.00	87
80400	40	16	45	2	2	.00	87
80520	52	13	34	1	1	.00	91
80550	55	15	41	1	1	.00	87
80580	58	16	55	2	2	.00	87
80640	64	16	53	2	2	.00	89
80660	66	12	28	1	1	.00	81
80730	73	15	51	1	1	.00	87
	76	12	29	1	1	.00	79
	77	14	40	1	1	.00	83
	80	11	40	3	3	.00	87
	84	15	35	1	1	.00	95
	88	14	35	1	1	.00	89
	89	14	39	1	1	.00	79
80960	96	12	31	1	1	.00	81
80970	97	16	52	3	3	.00	
81060	106	16	42	1	15	2.00	
81070	107	16	45	1	1	.00	
81100	110	16	52	1	1	.00	
81110	111	16	55	2	2	.00	
81210	121	16	43	1	1	.00	
81220	122	15	41	2	2	.00	
80280	128	15	39	1	1	.00	
81330	133	12	32	1	1	.00	
81450	145	10	38	1	1	.00	
81500	150	16	47	1	1	.00	
81510	151	14	36	1	1	.00	
81520	152	12	32	1	1	.00	
81650	165	14	46	2	2	.00	
81660	166	16	49	2	2	.00	
81670	167	16	44	1	1	.00	
81760	176	13	34	1	1	.00	
81850	185	13	43	1	1	.00	
81900	190	15	51	5	5	.00	
81940	194	13	43	6	6	.00	

ED031066

EF 003543

BOARD OF DIRECTORS

Milton C. Mumford, CHAIRMAN
Chairman of the Board and President, Lever Brothers Company

Alvin C. Eurich, VICE CHAIRMAN
President, Aspen Institute for Humanistic Studies

Clay P. Bedford, *President, Kaiser Aerospace & Electronics*

James C. Downs, Jr., *President, Real Estate Research Corporation*

Henry Dreyfuss, *Industrial Designer*

Morris Duane, *Attorney, Duane, Morris and Heckscher*

Harold B. Gores, *President, Educational Facilities Laboratories*

Frederick L. Hovde, *President, Purdue University*

J. E. Jonsson, *Chairman of the Board, Texas Instruments, Inc.*

Philip M. Klutznick, *Klutznick Enterprises*

J. L. Morrill, *Consultant, Ford Foundation*

Winthrop Rockefeller, *Winrock Farm, Morrilton, Arkansas*

Thomas J. Watson, Jr., *Chairman of the Board, International Business Machines Corporation*

Benjamin C. Willis, *General Superintendent of Schools, Chicago, Illinois*

OFFICERS

Harold B. Gores, *President*
Jonathan King, *Secretary and Treasurer*

STAFF

Lillian Butler, *Publications Associate*
Robert M. Dillon A.I.A., *Consultant*
Margaret Farmer, *Editorial Associate*
Ronald W. Haase A.I.A., *Architectural Associate*
Arnold J. Kuesel, *Assistant Treasurer*
James J. Morisseau, *Editorial Associate*
Bernard P. Spring A.I.A., *Consultant*
Ruth Weinstock, *Research Associate*

Educational Facilities Laboratories, Inc. is a nonprofit corporation established by the Ford Foundation in 1958, to help American schools and colleges with their physical problems by the encouragement of research and experimentation and the dissemination of knowledge regarding educational facilities.

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

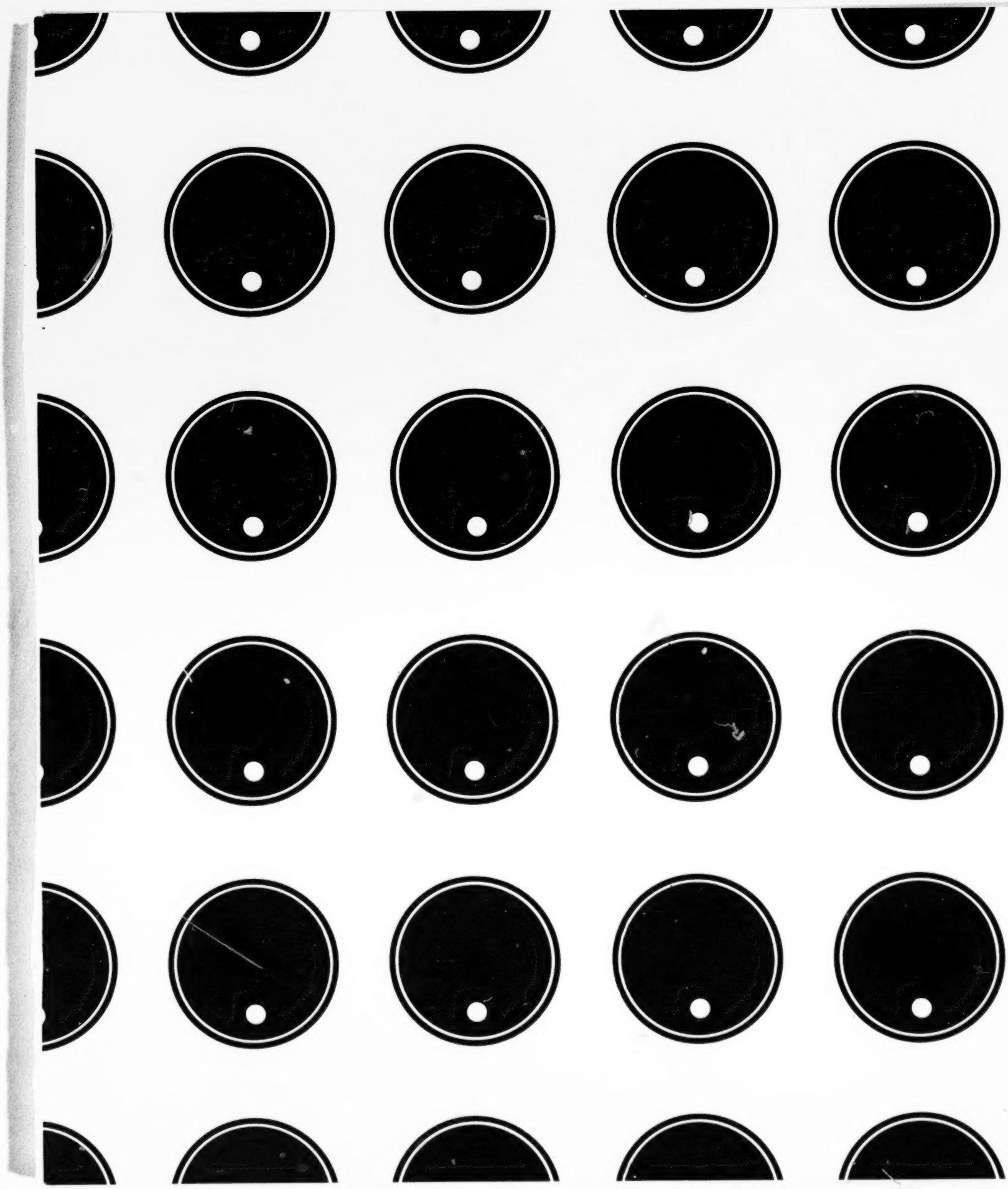
SCHOOL SCHEDULING BY COMPUTER [redacted] THE STORY OF GASP

REPORT BY JUDITH MURPHY

CENTER SECTION BY ROBERT SUTTER

"PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY Educational Facilities Labs., Inc.
TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE U.S. OFFICE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER."

© 1964 Educational Facilities Laboratories, Inc., Library of Congress Catalog Card No. 64–8927
Additional copies are available from Educational Facilities Laboratories, 477 Madison Avenue, New York 22, N.Y.
First Printing, December 1964; Second Printing, June 1966.



WHAT IS A MASTER SCHEDULE

(*variously known as the master program or the timetable*)?

Laymen may think they know, but school administrators have reached for high-flown metaphors in trying to convey its singular importance: "The master program is to the high school principal as the musical score is to the concert director, for in either case a soundly planned program, harmonious and tightly knit in all of its component parts, will determine the effectiveness of the individual and his organization." Another administrator has put it more starkly: "The schedule is in many cases the principal, if not the only, bulwark standing between the administrator and chaos."

Indeed, the importance of the master schedule can hardly be exaggerated. It abstracts, in words and numbers, the essence of the school. For a given school year, it sets forth in precise detail who is going to do what for every period of every day in the week. Subjects, students, instructors, classrooms are all assigned. From the close study of a master schedule, a canny reader can learn much of a school: the programs it offers; the constraint or freedom that affects students' choice of courses; the school's position on the spectrum that runs from ultraconservative to radical; its size, resources, shape—even its philosophy. In some schools, perhaps most schools, the schedule dominates the students and teachers it is presumably designed to serve. In a minority of schools, happily on the increase nowadays, the schedule reflects fresh efforts to enliven and individualize education.

Enter the Computer

Until recently, it was widely assumed that the actual building of a master schedule, involving so many variables and calling for so many administrative decisions, was far too complex and subtle for automation. Computers, to be sure, had proved themselves invaluable for all kinds of routine data processing in schools, but most people (including computer manufacturers themselves) discounted the feasibility of programing the intricacies of a master schedule. In the face of this general skepticism, experimenters nonetheless persisted in the belief that scheduling could be automated.

Today, the experimental returns are coming in. This report deals with one successful program which harnesses a large digital computer to school scheduling. The device is **G**eneralized **A**cademic Simulation **P**rograms, or GASP for short and for memorability.

It is important to stress at the outset that the task under consideration here is the building of a *school schedule*—not simply the assignment of students to classes, or "sectioning," as it is sometimes called. For some years now schools have been using large computers for the latter task. But automated student-assignment programs, such as IBM's CLASS have been applied to the old-fashioned, handmade, school schedules.[1]

GASP has now demonstrated that automation can go beyond sectioning and actually produce the master schedule itself. Not only does the program perform faster and more efficiently than the most ingenious and tireless schoolman, but—much more important—it produces a schedule that takes fuller account of student and teacher preferences, of innovations like team teaching that complicate schedule-making, and of almost any array of circumstances peculiar to the school in question. GASP's value is demonstrable, above all, for schools introducing new practices. Furthermore, the program provides a powerful tool toward better and more economical school design: GASP can simulate the academic operation of schools not yet built or even designed and thereby enable administrators to plan for optimum utilization of physical plant, avoid costly mistakes, and save millions of dollars. As we shall see later, revised plans based on a $15,000 GASP study may save more than $10,000,000 in construction costs for three projected junior-college campuses in St. Louis.

1. *For further information on computerized scheduling and sectioning programs, see Appendix A.*

It goes without saying (but it's worth saying again, if the confusion endemic to computerdom is to be minimized) that GASP produces good results in direct ratio to the thoughtful collaboration of its human partners. Unlike routine data processing by computer, the success of GASP presumes interaction between scheduler and computer. And, as Robert Holz, the young creator of GASP points out, "the impossible will remain impossible even with computers." It may be significant, however, that Mr. Holz and his even more youthful associates at MIT—while they make a point of the fact that they never refer to their IBM 7094 as "him" or "her"—regularly say "we" do this and "we" do that when referring to the convolutions of GASP within the circuitry of the big computer.

Scheduling Old Style

Before getting into the details of automated scheduling, it will be useful to look more closely at the process of building a master schedule by hand. As every conscientious principal knows, it is a frustrating, tedious, and time-consuming job. It is easiest in the traditional school that operates on set time periods with an immemorial roster of subjects. There are some schools of this type that scarcely vary their schedules from one year to another. It is easier to build a schedule for a very large school than for a small one; even if such a school groups its students by ability or "track," courses are apt to be divided into so many fungible sections that conflicts are readily avoided. Not so in the small school—even the fairly conventional one—where many, if not most, courses may be limited to a single section.

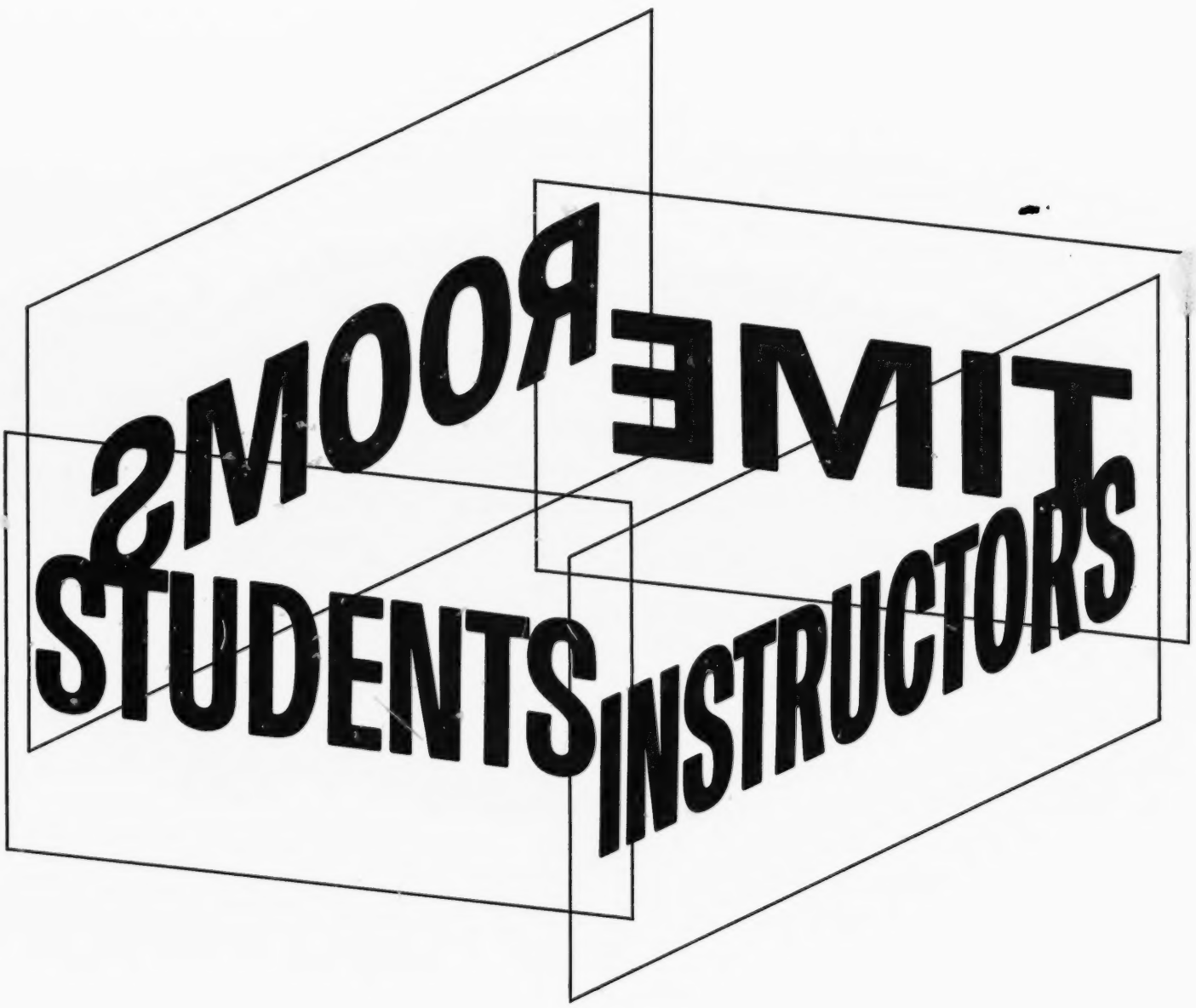
The task is difficult, nonetheless, for the principal of any kind of school who tries to revise and adjust a schedule to match school goals, changing needs, resources of plant and staff, and student capabilities and wishes. The vast majority of schools, whatever their size, build their schedules by hand. Only very big schools, with enrollments of 5,000, perhaps, or more (a small fraction of all U.S. high schools), employ professional schedulers. In most schools, and in virtually all small high schools, the job falls to the principal or assistant principal, with varying amounts of extra help. Nearly always he will have considerable clerical help—his own office staff at least. A well-heeled

school may hire a few months of *expertise*—perhaps a math teacher (or even two) who works through the summer under the principal's direction.

Whatever the arrangements, an undue amount of professional time and talent goes into schedule construction. It is true that a good part of the job is sheer clerical drudgery: the listing of subjects, rooms, instructors; the tallying of student choices; and the making of a conflict chart. Much of this work can be done by competent office help; moreover, in recent years, many schools have semi-automated part of the job, through key- or card-sorting. The time-consuming trouble comes with the next stage of the job: the actual construction of a timetable out of all these elements. What remains for the principal or his surrogate is to sort out and try to match these elements, in the best possible pattern, avoiding conflicts or resolving them, for the greatest good for the greatest number of students and teachers. Untold hours go into hanging little, round, metal-edged tags (representing sections) on hooks set into a big board (each hook representing a room).

The ultimate job involves much that is merely clerical, to be sure: the trouble is that clerical detail is so intermingled with administrative judgment that little delegation of the work is possible. A clerk could juggle all the data around and pursue the trial-and-error strategy the task demands; but it takes a deeper knowledge of school and staff to balance conflicting demands and resolve potential time conflicts—to decide what to do, for instance, with this set of facts: Teacher A is better than Teacher B, A is available for X period, B is available for Y, student conflicts are minimized in Y period.

The skeptical reader may well have pounced on the foregoing sentence as clear vindication of his basic doubts and fears about computers. How can we expect an IBM 7094 to display the subtle knowledge of a school and the balancing of human factors that we don't expect in a clerk? How can a computer know that ambitious Teacher C is taking courses at the university and can teach no classes after two o'clock; or that veteran Teacher D has what amounts to a lien on Room 302; or that Typing III should be scheduled early in the day since so many students in this course are released for afternoon jobs? ANSWER: the computer obviously can't know these and all the other subtleties that go into a good schedule. But "we" can tell it to take account of them, through adaptations in the basic GASP programs and through the way the



The 4 dimensions of scheduling as a mathematical problem.

school's own data is prepared for the computer. In anything so complex as scheduling, it would, of course, be impossible to include instructions covering every alternative and every local quirk. But the computer can store a large number of these "parameters" (to use the programing term) in its memory and apply them where needed. Properly programed, the computer behaves like a supremely efficient clerk that is on top of all relevant data and never forgets for an instant the school's operating imperatives and *caveats*.

School Reforms and Scheduling

If schedule construction is a formidable task in the conventional school that changes little from year to year, it looms like a monster to the principal of a school embarked on educational innovation. Consider the potential scheduling difficulties inherent in team teaching, for instance. Or in non-graded plans where individual students progress at their own rate through the school. Or in the redistribution of standard classes into large, medium, and small groups. Or in provision for independent study and honors work, or wide-ranging electives. Or in the strict application of ability grouping, subject by subject. Or in such innovations in the school day as modular scheduling, or flexible periods.

During the past decade, high schools the country over have tried to improve their educational offerings in many ways—by faithfully following the Conant prescription, by adopting the new curriculum revisions, by installing advanced placement and other honors courses, by experimenting with foundation-subsidized forays into team teaching, teaching by television, flexible arrangements of time and students.

The results are not overwhelming, but they are ponderable—ponderable enough for the colleges to feel their impact and appreciate the need to bolster their Freshman courses. The quiet revolution now under way in American education seems destined, in part, to remold secondary education—to make it more sensitively attuned to the provision for individual differences so much touted, so little achieved, by twentieth century educators. One result will be increasingly complex high-school programs, "comprehensive" in fact as well as in name. And to the extent that high schools truly provide instruction calibrated to the range of talents and abilities of their students, they will

have to operate on a programmatic basis much more like college than like the standard high school of today and yesterday.

Which brings us to a compelling statistic. Up till now, American colleges have not had to worry much about efficiency. With space and professors to spare, they have maundered along their old, accustomed paths, vaguely equating "liberal" with "laissez-faire." They could comfortably accommodate the modest increase of students each year with no harsh revision of their vaunted low student-professor ratios, hoary departmental logrolling, and relaxed scheduling with nearly as many individual programs as students. Now, of course, the deluge is upon them: the baby boom of the forties has hit higher education. But that is another story.[2] The point to be made here is that if the colleges are to save their central traditions, which surely include programs custom-tailored to the individual students, they must be willing to slough off many peripheral habits that are trivial and wasteful. By the same token, insofar as secondary schools aspire to truly individualized instruction and programs, they face problems comparable to those of the colleges. But if our secondary school facilities were as poorly utilized as the colleges are now, the United States would need something on the order of *twice* the present secondary school facilities. The average college today makes use of its instructional space roughly 40 per cent of the time during its operating hours, and even then only 50 per cent of the available seats per class are filled with students. By contrast, the average high school makes use of 80 per cent of both rooms and seats during the school day.

In short, high school reform mandates the greatest possible utilization of existing and future facilities. And here GASP (and comparable or related programs) comes in strong. For, thanks to the adaptability of the program itself and the incredible speed and accuracy of the big computers, a school can build a successful schedule to provide for all kinds of innovations without sacrificing efficient utilization of staff and facilities. To make by hand a master schedule for a conventional high school takes from one to two man-hours per student—or upwards of 1,000 man-hours for a school of medium size, including a high percentage of expensive and scarce administrative time. To make

2. See Bricks and Mortarboards, a *1964 report from Educational Facilities Laboratories on facilities for colleges and universities.*

by hand a schedule for an innovating school program—with unconventional time periods, say, or school-wide team teaching—is likely to take at least twice as long. One GASP user, the principal of a new and highly unconventional high school, says flatly that to schedule his intricate program by hand is a practical impossibility. "The money and energy and time demanded by a handmade schedule means we either automate or we go out of business" (i.e., revert to a conventional school program).

Four Schools Automate Their Scheduling

GASP evolved from a problem that faced MIT in the late 1950's. Miss Mary Hurley, who had been in charge of MIT's scheduling for nearly three decades, was due to retire in a few years. With her would go the virtually uncommunicable *expertise* of scheduling this huge, complex university. Miss Hurley's impending retirement brought to a head scheduling problems that had been growing increasingly difficult to solve by hand, as new demands by faculty and students produced more and more challenges to conventional timetable construction.

Robert Hewes, who had become registrar in 1956, saw salvation in the computer. He organized a number of exploratory sessions with experts in an effort to achieve a "bold new look" in scheduling, and decided to assign Robert Holz, who had been graduated from MIT in 1959, to work full time on the project. The computer program that became GASP was initially devised to schedule MIT, as it now does. By 1963, Mr. Holz (working on an EFL grant since 1961) felt that GASP had been sufficiently debugged to be offered to other institutions. Meantime, Mr. Hewes (now MIT's Director of Institutional Studies), Mr. Holz, and another colleague had formed the private firm of Hewes, Holz and Willard to act as consultants in educational data processing. Among other things, the firm advises schools in the use of GASP.

To date, four high schools have experimented with GASP to build their master schedules and assign students. Three high schools operated with GASP schedules in the school year 1963–64, and built GASP schedules for 1964–65. They are Wayland High School, in Wayland, Massachusetts; Ridgewood High School, in Norridge, Illinois; and Cohasset High School, in Cohasset, Massachusetts. The fourth school, the brand-new $3 million Pascack Hills

High School, in Montvale, New Jersey, opened in September of 1964 with a program scheduled with GASP's aid. The schedule-building in these schools (except for Cohasset) has been supported by EFL. IBM very generously provided Wayland and Ridgewood with computer time for testing and feasibility studies. Next year Hewes, Holz and Willard will schedule a number of additional schools besides Cohasset on a commercial basis: two high schools in Quincy, Massachusetts; Centennial Joint Schools, in Bucks County, Pennsylvania; and a high school in Charleston, West Virginia. EFL no longer considers the program experimental.

The three schools that have been using GASP schedules for a year report general approval, and even enthusiasm, for the experiment. The schools are quite diverse—in population, size, plant, and program. They are alike in their deviation from the humdrum and in their determination to achieve high instructional quality. Wayland and Ridgewood are both "Trump" schools—which, as every up-to-date schoolman knows, is shorthand for schools embodying all or most of the innovations propounded by the Commission chaired by J. Lloyd Trump in the extensive project undertaken for the National Association of Secondary-School Principals, with foundation backing.[3] Thus, for instance, Wayland and Ridgewood have introduced school-wide team teaching, and both have organized most instruction into flexible groupings of various sizes. Cohasset offers a more traditional high-school program, with an important difference: the school operates five tracks, or ability groupings, in each basic subject, and assigns students independently in each subject.

Mr. Holz and his associates began working with Ridgewood High in the fall of 1962. The school had opened two years before, in an industrial suburb of Chicago which had previously sent its high school students to a neighboring district. It opened as a full-blown "Trump" school, with a brand-new staff and 425 freshmen and sophomores. Not until the school year of 1962–63 did Ridgewood have its full four-grade complement enrolling 950 students. Today there are just over 1,100.

Ridgewood's educational program is enough to stagger any scheduler. All instruction is divided into 1) one large group presentation, 2) discussion (or seminar) groups, 3) laboratory groups (where students can work on their own

3. *J. Lloyd Trump and Dorsey Baynham,* Focus on Change: Guide to Better Schools (*Chicago: Rand McNally & Company, 1961*).

with their teacher handy for consultation), and 4) study and project groups. Ridgewood has dispensed with the conventional class of 25 or 30. Team teaching prevails throughout the school.

On top of these innovations, Ridgewood has operated from the start on a modular schedule, with the school day divided into 20 modules of 20 minutes each. Instructional periods range from 2 to 4 modules. Students are passing every 20 minutes. No bells ring. A system of synchronized clocks enables teachers to dismiss their groups promptly and quietly. Insofar as scheduling permits, Ridgewood tries to follow each teaching team's recommendation as to the best mix of large-medium-small instruction. The school allows students a broad choice of electives. Roughly half of Ridgewood's students have honors passes, which permit them to arrange their own study time. Furthermore, unlike the conventional high-school schedule that basically repeats itself for the five days of the week, the Ridgewood schedule is, in effect, re-cycled every day, with some classes meeting Monday and Thursday, some Tuesday and Friday, and others Wednesday. The result is a more complex schedule, analogous to the college pattern.

During the school's first few years of operation, Ridgewood's principal, Eugene Howard, plus several assistants, built the schedule by hand. With only Freshman and Sophomore classes to start with and plenty of space, a manual schedule was possible, but just barely so. In the words of C. F. Malmberg, now in charge of Ridgewood's scheduling: "The complex program presented terrible problems of conflict and balance." The schedulers were constantly being faced with dilemmas like 50 students in a class designed for 15. In Ridgewood's third year—its first as a full four-year high school—they solved part of their problem by using IBM's CLASS for student assignment, and began working with Mr. Holz on plans for a GASP-produced schedule the following year. Mr. Malmberg says that if Ridgewood High were to try to build a master schedule by hand today, with over 1,000 students enrolled, it would take the full-time attention of a faculty member and a clerk from February until September, plus another full-time professional from June on. And it still wouldn't be a really good schedule, he adds.

Now, with two years of experience behind them, the Ridgewood people are making effective use of GASP to build a generally satisfactory schedule

that takes substantial account of their most important educational designs. They prepare all their own data, and are now so adept that coding and punching the cards takes no more than three weeks of clerical time. Ridgewood's first GASP schedule required about 20 runs—some 10 hours of 7094 time; the schedule in 1964–65 took about the same time. Much of the computer time used on these schedules was for program research and development and will not have to be repeated. Other special Ridgewood factors added time, such as special programing for lunch and study assignments and the novel demands of the modular time periods.

According to Principal Howard, Ridgewood's experience with GASP has demonstrated that:

1. *A schedule of great complexity, such as the Ridgewood schedule, can be built by computer at less over-all cost than if it were done by hand by an administrator.*

2. *The computer-built schedule has fewer conflicts than does the handmade schedule.*

3. *Class lists, room utilization lists, teacher schedules, and student schedules are extremely accurate. For a modular schedule such lists are almost impossible to develop accurately by hand except at great cost in time and money.*

4. *The greatest advantage to the school of a computer-built modular schedule is that the scheduler, in the process of generating his master schedule, is able to construct a large number of preliminary schedules. He can analyze each and then incorporate improvements in each succeeding run until he reaches a satisfactory and workable combination of courses, time allocations, teachers, and rooms within the scope the school has indicated.*

Hewes, Holz and Willard began working with Wayland High School in the spring of 1963. Wayland, Massachusetts, is an upper-middle-class suburb west of Boston, with strong academic ambitions. In 1960, its high school had moved into a new building designed to accommodate a drastically revised educational program that had been under study and development since 1958. In two years, Wayland High had changed from a wholly traditional school into a showcase experimental school with team teaching across the board and instruction organized into groups of various sizes.

While Ridgewood's building was modified in mid-design for the team teaching program, Wayland's was designed from the start to fit a well-articulated program of team teaching. Ridgewood High occupies a large, fairly conventional, two-story building. Wayland, on the other hand, is laid out in campus style, with separate buildings for cafeteria and administration, languages, social studies and business, mathematics and science, arts center, and a field house for physical education.[4]

Unlike Ridgewood, Wayland operates on a standard school day of seven periods. The school allows its 800 students wide elective choices (though an administration new in 1963 has somewhat moderated this policy). One special Wayland policy with intricate implications for schedule-building was the school's original requirement that each student pursue each basic subject in one large meeting, two medium-sized meetings, and one small meeting per week. (By fall 1964 the school was making departures from the 1-2-1 pattern with departments electing other patterns such as 1-3-1 or 0-3-1.) Further, the school required the assignment of each student to a *particular* medium group breaking off from his assigned large group, and to a *particular* small group breaking off from the medium. In practice, the small-medium tie sometimes has to be broken when it presents insoluble scheduling conflicts. Similarly, the constraint imposed by trying to schedule each subject into its designated building sometimes has to be broken to make a workable schedule.

What a program like this means is that out of a school population of 800, you may have something in the order of 600 or 700 different individual student schedules (with the revised policy on electives, Wayland's new principal, Ray Hettler, figures that this figure has been cut to around 350). Wayland's initial experiment with GASP—the master schedule for 1963–64—took 20 runs (about six hours of 7094 time) and entailed an unusual degree of adjustment and arbitration between runs. But in the end GASP helped to produce a basically satisfactory schedule, and Wayland opened in 1963 with far fewer conflicts and problems than in the preceding five years of handmade schedules. "The whole thing was smoother," says Mr. Hettler, "we had far fewer unhappy kids and parents." This year, Wayland's Superintendent R. Bruce McGill reported, "I was delighted with our opening of school and can report . . . that the professional staff, students, and comments from parents indicate

4. For further information see High Schools 1962, a report from Educational Facilities Laboratories.

that it is the smoothest start in some seven years. It does prove that, even with as complex a schedule as ours, a good master schedule can be produced which centers around the elections of students."

In times past, the scheduling process has been a summer-long, full-time headache for three staff members, including the former principal, the present assistant principal, and mathematics teacher Richard Randlett (who is now associated with Mr. Holz at MIT). Mr. Randlett speaks with feeling of those long, hot summers with the Royal McBee sorting cards and the long pins (1960)—with the Remington Rand keypunch and sorter (1961)—and with the IBM keypunch and sorter (1962). Inquiries about the possibility of computerizing the master schedule, says Mr. Randlett, encountered either blank looks or detailed and plausible objections. One year, he recalls, he and his colleagues worked straight through from eight o'clock one Friday morning to four o'clock Saturday afternoon to complete the schedule in time for opening day. With GASP, Wayland requires about two months' professional time, plus about the same amount of clerical time for data preparation.

Mr. Hettler, as noted, reports general satisfaction with GASP as a means to a better schedule and also as an efficient means of testing alternatives and future plans. But in so doing he reflects that humanistic uneasiness with computerdom mentioned earlier in this report. "The educator really isn't geared to this program," he says. "Scheduling is essentially a mathematical, not an educational, problem. To use GASP well requires someone with the proper know-how, an expert." And yet, he suggests, the educator feels somehow uncomfortable in relinquishing a measure of control of the schedule to a computer or even a computer man: there is instinctive objection to an arrangement that puts something the educator doesn't fully understand or control between him and his schedule. At the same time, and most interestingly, Wayland's principal exemplifies an important point that Mr. Holz makes about GASP in citing the program's ability to reduce the clerical load of the scheduler and save him for the difficult decisions. Says Mr. Holz: "Use of such a program means that the scheduler may know less about a particular schedule in detail. But he can learn more about his scheduling problem in the larger sense. Used intelligently, a program such as GASP is sure to give the scheduler insight into his future needs and problems."

So Mr. Hettler—in discussing Wayland's difficulties with scheduling wide student electives; trying to maintain the ties between large and medium, medium and small groups; trying to provide music or band within the school day—reaches the following reflection: "Maybe the computer in trying to resolve unresolvable conflicts is trying to get across a larger message. Maybe it's telling us that modern education requires a different kind of school day—more like the college day.

"It's possible we simply can't run the kind of educational program we want in the standard 8:30 to 3:00 school day. Why not start at 8:00, with half the student body—have the rest come in later and stay till 5:00, say?

"We'd have more room in the school, more time and space for student electives—for students to combine courses in many ways—for students to use labs on their own—for teachers to put into effect new ideas.

"Sure, with an 8:00 to 5:00 school day, and a staggered program, we'd have to run more buses. But mightn't that be cheaper than putting up more buildings?

"Maybe the computer is giving us useful clues to the school of the future. We educators shouldn't be so reluctant to experiment, so frightened of change, so concerned with measuring up to the old patterns and expectations."

In general, the GASP experiments at Ridgewood and Wayland have turned out well. Originally, both schools planned to build schedules by hand in the traditional way at the same time they produced the GASP-built schedules, so as to make possible a solid comparison of the two. But both schools, after seeing the results of the first few computer runs, gave up the idea of making all that nonessential work for themselves. This, together with the absence of hard data on previous scheduling at these schools, makes comparison of computerized and manual schedules difficult, as Robert Holz has pointed out. He goes on to say: "The people closest to the scheduling process at both Ridgewood and Wayland maintain that fewer student conflicts arose, that there were fewer compromises on what could be considered 'ideal' scheduling policies of the schools, that the opening of school was smoother than in former years, and so on. These last claims must not be construed to mean that scheduling was 'perfect' in any sense. There were still student conflicts, there was still scheduling difficulty when space utilization figures exceeded

80–85 per cent in a given area, there was still unhappiness on the part of some teachers who could not be accommodated with their favorite classes at their favorite times."

And he adds, with true academic diffidence: "The claim is only that some improvement resulted in applying data-processing equipment to the schedule problem." An accurate, if over-modest, evaluation.

No such diffidence marks the reactions to computerized scheduling of Anthony D'Antuono, principal of Cohasset High School. Mr. D'Antuono couldn't be more pleased with GASP if he had invented it himself. Indeed, he has a certain proprietary feeling about the program. For five years or so, as he struggled each year from January to July handmaking his complex schedule, he had been harboring the notion that the job could surely be automated. But over the years, whenever he made inquiries at data-processing centers and other likely spots, he would get pretty much the same answer: "Possible, yes. But the cost would be prohibitive." Then early in 1963 on the eve of his annual tussle with the schedule, Mr. D'Antuono tried once more. This time IBM put him on to Robert Holz. Before long, to Principal D'Antuono's vast satisfaction, Cohasset had its first GASP-produced schedule.

Cohasset is a prosperous suburb on Boston's South Shore, and its trim brick, six-year high school, standing on a knoll at the end of an oak-lined drive, is the very protoype of the American dream school dear to magazine-cover artists. Inside, all is sparkle and good cheer—and a remarkably stiff academic program in which the principal takes enormous pride. The entire student body is grouped homogeneously in each subject area "according to mental ability, prior academic record, standardized test results, and teacher recommendation." This strict homogeneous grouping subject by subject, in as many as five groups, is the fulcrum of Cohasset's educational program; it makes the task of scheduling 650 students tantamount to preparing 650 individual schedules. The school's small size aggravates the difficulties, since there are few, if any, interchangeable sections.

For years, Mr. D'Antuono evolved his master schedule by the long-drawn-out, tortuous process of building a conflict chart by trial and error. In recent years he had made use of the keysort system which he found helpful in working out the conflict chart and in tabulating data. But there was still the main

job of the timetable, which had to be put together by hand. And Mr. D'Antuono was conscious of the fact that his handmade schedule, while ultimately workable, was only a rough approximation of what Cohasset needed. "A human being," says Mr. D'Antuono, "is simply incapable of exploring all possible solutions in this mass of variables."

The use of the computer, on the other hand, assures Mr. D'Antuono that a much wider array of alternatives have been considered and that the result is not only better than his old handmade schedule, but actually "more human." On this point he is quite vehement, since it touches the major objection to computerized scheduling that he is always meeting from his fellow principals. According to Mr. D'Antuono, the scheduler working with the computer can take much fuller account of the human element than any scheduler working alone. "When I did the schedule alone," he says, "I could only scan a handful of possibilities and then do what I thought best. The computer analyzes hundreds of possibilities, and the students benefit." Furthermore, the time saved by GASP gives a principal more time for the purely educational and supervisory aspects of his job.

Mr. D'Antuono says that he spends no more than a day preparing the input for GASP: a list of teacher assignments, the course selection of the students in the assigned groups, a list of the subjects to be taught (with the number of homogeneous groupings in each area), a list of classrooms, and a list of time-patterns for each subject (e.g., five days a week, twice a week, one double period for laboratory, etc.). All this material can be tabulated on two large ledger sheets. The basic data, of course, has already been gathered by his staff—but this they would have done anyhow. And D'Antuono, with his intimate knowledge of the school, of teacher preferences, and so on, simply dictates the requisite data to his secretary.

Cohasset's first GASP schedule was produced in 1963 for the 1963–64 school year and covered some 400 students in the senior high school. It took seven runs of the 7094, or a little over half an hour of computer time. Mr. D'Antuono spent a total of perhaps two and a half days working with the computer people, analyzing the output between runs, and resolving conflicts by hand. He feels now that many of the flaws in the early runs came from inexperience with GASP and faulty preparation of data. Cohasset was scheduled

more quickly this year (even though GASP scheduled all six grades). In any event, the seventh run produced "the best schedule ever," with few conflicts, a good match of student to group, and no need to reassign a single teacher. Mr. D'Antuono summarizes the experiment: "What once had taken months to accomplish can now be achieved in a relatively short period of time. It is now possible, through the use of the IBM 7094 computer, to construct an entire school program, regardless of its intricacies and complexities, within a minimal span of time and with an optimum of efficiency."

Pascack Hills High School in Montvale, New Jersey, opened its new building this fall with a transitional program—partly conventional and partly team teaching. For months before the school opened, Principal Donald Wyckoff experimented with the GASP program, working with a senior program analyst from International Telephone and Telegraph's Data Information System Division, in nearby Paramus. He was aided by his own IBM-trained, part-time assistant. The school—opened with between 700 and 800 freshmen, sophomores, and juniors—introduced such innovations as team teaching, individual study programs for both advanced students and slow ones, audit and remedial classes (the latter, for example, for "disabled readers") and nongraded classes in areas like foreign languages (where many students in this sophisticated community may be competent well beyond their age).

Pascack Hills school officials mean to introduce changes gradually and with minimum fanfare. For one thing, parents, anxious about getting their children into college, are actively interested in the school program and quick to question any changes. The school wants the chance to experiment freely without arousing needless concern among students, parents, or staff. Principal Wyckoff and Superintendent James McNeil, accordingly, exploited the trial-and-error possibilities of GASP in the spring and summer before the new school opened. Pascack Hills required eight runs to build its schedule and section its students. Only 54 conflicts arose and only one was hard to solve. There were a few problems with teachers' schedules which could easily be solved next time around.

[redacted]

GASP IN YOUR SCHOOL

By way of making clear how GASP works, what it can and what it can not do, let us suppose that you are the new principal of a suburban high school with around 1,000 students. Aside from a couple of advanced placement classes and a language laboratory, the school's program is essentially the same as it was a generation ago. Now change is in the air. The educational ferment of the sixties has reached your town. School board meetings and PTA meetings are punctuated with talk of team teaching, programed instruction, the possibilities for independent study, and more sensitive groupings to meet the needs of all the students, from the slowest to the most gifted. As principal, you are eager to introduce innovations. An added impetus is the prospect of moving in a few years out of the overcrowded old building into a new high school. You would like to have an improved and updated school program dictate the design for the new building.

With the approval of superintendent and school board you can proceed with pilot experiments to be put into effect a year hence—teaching teams in English; new approaches in mathematics; an experimental redistribution of social science classes into large, medium, and small groups, with trials of independent study and televised instruction.

School committees work through the fall and winter to plan these complex changes. As plans proceed, you realize with increasing force that the tedious old job of schedule-building will be a real killer this time around. In

February, you get authorization for extra clerical help and for the release of a staff member to work full time on the schedule. In June, you add another teacher, and the three of you, plus two clerks and a half, work on the schedule clear through the summer. By September, you have a workable, if far from satisfactory, schedule. Conflicts still persist on opening day, and you and your aides are ironing them out a month after school starts. You are confronted, let's say, with students who had elected a subject on the assumption (which turned out to be wrong) that they'd pass the prerequisite during summer school.

By the following year, you have made arrangements to simplify the schedule problem by delegating the student-assignment phase to a computer. But the more onerous task of building the timetable—which now, with further innovations, will be even worse—must still be conquered by brute force, even if sectioning is automated.

Then, along about March, you pick up an interesting bit of news shoptalking with your peers at a regional meeting. You learn about a package of programs called GASP that a number of high schools have used with considerable success. After talking to the people at MIT and checking with the schools that have used the program, you are convinced of GASP's potential contribution to your scheduling problems and thus to the school's educational innovations. You and your superintendent succeed in selling the idea to the school board, which appropriates $5,000 for the experiment (including the cost of expert services). It seems likely that the total cost will run below this figure, but, because of various imponderables, it seems wise to overestimate rather than the reverse.[5] There is considerable evidence that GASP, whatever its out-of-budget expense, can effect a true economy in its saving of expensive professional time, including your own. But you recognize that, from the taxpayers' point of view, it is a budget extra.

A nearby university, as it happens, may wish to grant your school the three hours or so of 7094 time you will probably need for GASP, together with the estimated 20 to 30 hours the program will need on the university's 1401 for all the output you want. Your experienced scheduling assistant goes to work at the university data-processing center with the consultant you have retained.

5. *See Appendix B for details on GASP costs.*

Strictly a noncomputer man yourself, you recall your initial reaction to the GASP manual, a volume calculated to strike panic into any humanist's heart. It is a blockbuster as thick as a telephone book, and twice as wide. The text begins disarmingly enough (once you have hastily skipped the seven-page table of contents), describing the program and its purposes in perfectly acceptable English. What unnerved you was the manual's resort, not to *machine* language (though it includes this, too, and by now you've accepted the fact that the 7094 understands only 0's and 1's), nor even to *programming* language (though there's plenty of this, and you're now prepared to believe that the 7094 can read "CLA" as "clear and add," and translate it into binary digits, or "bits"). What really gave you pause was the manual's resort, as it gained momentum in describing GASP, to language which was obviously English but which could have been Jabberwocky or Bantu for all it penetrated to your understanding. Bemused, as you leafed through those outsize pages you merely registered the existence of something called an octal breakpoint dump, the viability of legal pseudo-operations, the importance of distinguishing between them and something called macro arguments, the need to assemble your symbolic with care and not to encrunch your END cards, and the role of the Boolean or.

Your dismay stemmed, of course, from your first brush with programer (as against programming) English, the language which computer men use to communicate with each other, and not necessarily to confound the uninitiated. It turns out, happily, that there is no need for the principal to understand, let alone master, the manual, and—further—that a little concrete experience with a computer program elucidates much that was abstractly dark. Your assistant (who has some data-processing experience) finds his way around the manual as well as he needs to before long.

It has been widely recognized, in recent years, that educational innovations —such as team teaching, teachers' aides, programed instruction, television— succeed in improving instruction to the extent that they reflect careful, thoughtful, integrated planning. Applied as stylish gimmicks or imposed by administrative fiat, they are apt to contribute little to a school beyond ready publicity in the local paper.

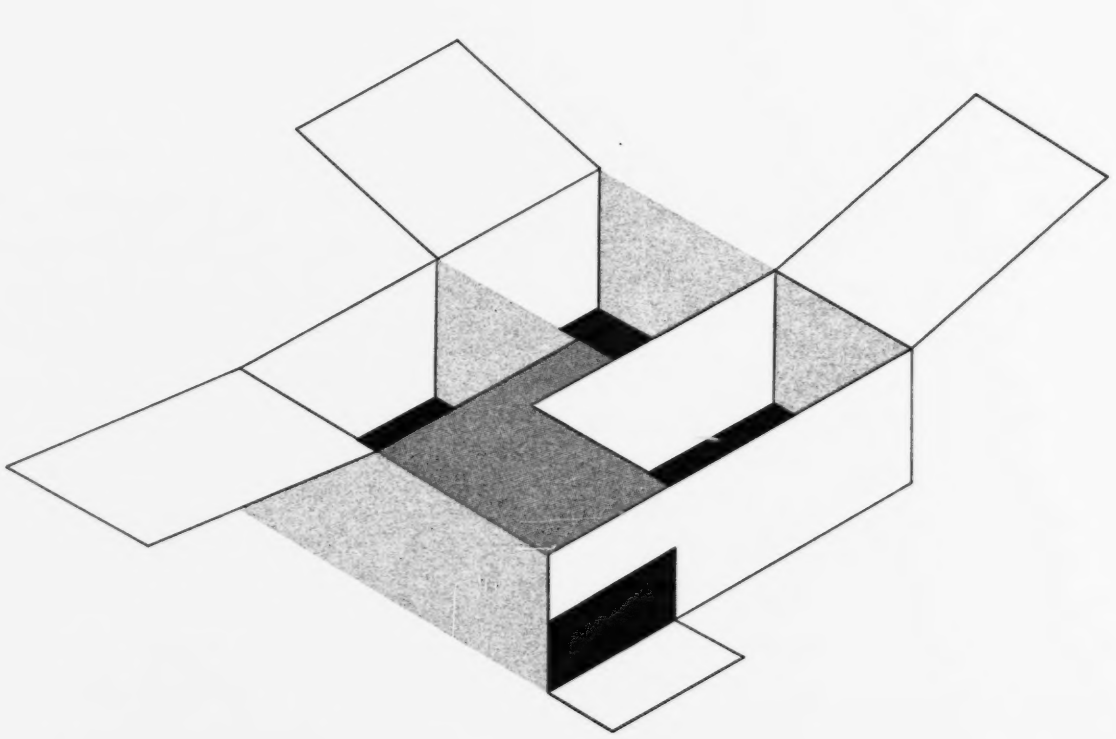
So with GASP, as you soon discover: it can provide major help in carrying out well-laid plans, and contribute mechanistic patience and accuracy to

scheduling a school program that has been carefully designed to reflect administrative policy, recommendations by teachers and guidance staff, student and parent choices. But if the school dreams up a program that won't fit into the existing plant, the computer won't create extra rooms (though it can improve the use to which you put the ones you have). You can't expect the computer to schedule eight classes into a seven-period day. The computer unaided can't solve such chronic problems as how to cram band practice or athletics or an excess of electives into an already crowded schedule. It is essential, in short, for the *human* scheduler—the principal, or his assistant— to make important decisions before feeding data into the computer.

You and your staff feel reasonably confident that you have reached agreement on major decisions, that your program reflects a clear policy on innovations to be undertaken or continued, and that you have taken proper account of student requests and instructors' wishes. The next step, then, is to prepare the input for GASP. In general, this consists of the same material that the principal must bring together before he builds a master schedule by hand. It includes the following:

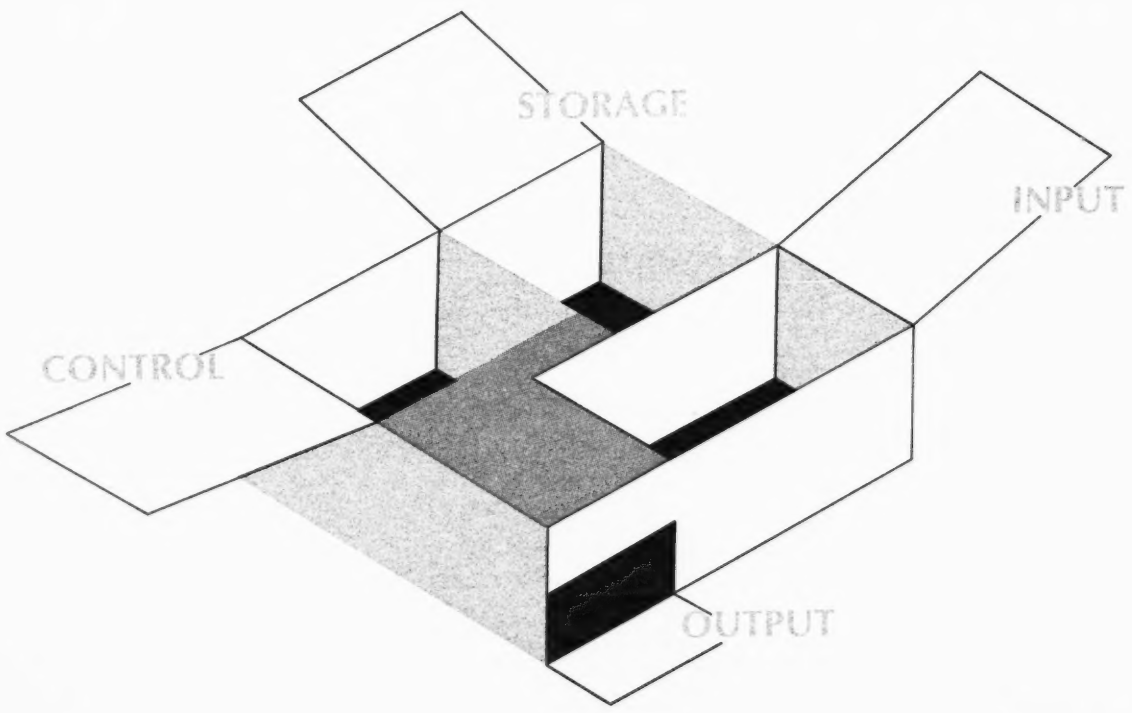
1. *A list of all instructors, by name and subject area.*
2. *A list of all rooms, giving capacity and special purpose, if any.*
3. *A list of all students and subject requests (or a sampling thereof).*
4. *Permissible time patterns (e.g., Monday—Friday first period, Tuesday fourth and fifth periods, etc.).*
5. *A list of all subjects, and for each one: the number of sections indicated, and all other relevant information, such as the maximum number of students per class, the type of time pattern for each subject (e.g., "five days a week," "twice a week," etc.), instructors available for each subject (and any preferences), and the distribution of rooms available for each subject (e.g., for English the list might specify any standard classroom, but for physics laboratory periods "physics laboratory only").*

A computer is comparable, in some respects, to an empty box waiting to be filled. Despite its complex components in all their electronic glory, the computer cannot function until somebody gives it a program and related data that human beings have prepared in special ways.



Data processing is a short way of describing what computers do —which is to go through a prescribed series of operations (the program) upon specific information (the data) to achieve a desired result. Computers come in different sizes and degrees of complexity, but they all consist of four basic elements: control, storage, input (program-plus-data), and output.

Whatever a computer does is determined by the program, which in effect transforms the computer into a special purpose machine.

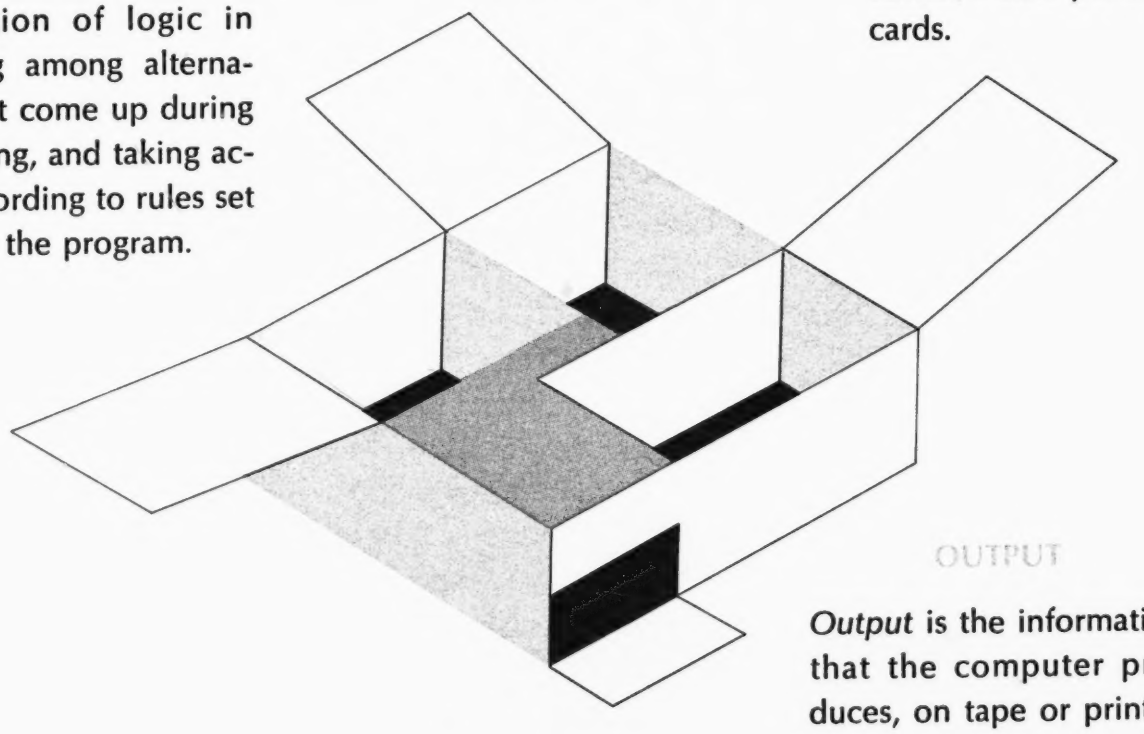


CONTROL

Control, which the program directs, coordinates the other parts of the computer to work together as a single purpose machine. Control performs such operations as adding, subtracting, multiplying, dividing, transferring, comparing, and storing. These operations, which produce guaranteed results, are called algorithmic. Control also applies heuristic operations—i.e., the application of logic in choosing among alternatives that come up during processing, and taking action according to rules set forth by the program.

STORAGE

Storage is somewhat like an electronic filing cabinet indexed to be instantly available to the computer. The program is filed in storage, and all data must be filed there before processing. The computer may rearrange stored data, or use it to calculate new data, storing the results.



INPUT

Input is, ordinarily, the program itself, plus unprocessed (raw) data, recorded in such a way as to be readily available to the computer. Input for GASP, for instance, is the program recorded on tape and the school's data punched on cards.

OUTPUT

Output is the information that the computer produces, on tape or printed on paper.

A computer specialist, a *Programer,* writes the program that will make the machine follow the proper series of operations on the particular data supplied. For instance, the GASP program tells the computer how to operate on the problem of juggling time, students, teachers, rooms, and subjects to produce a complete schedule. The school administrator must supply the detailed data about his school. GASP tells the electronic guts of the computer what to do with the data. As a school administrator, you originate the data. You get the program from a program specialist.

The computer must ingest data in a form of notation compatible with its electronic innards. This form of notation, called binary, is tedious for anything but computers. Happily for the human beings involved, computers can translate special code-languages into binary notation. Thus programers are spared the drudgery of writing lengthy programs directly into notation. These code-languages are akin to English but are limited in structure by the abilities of the computer to store information in binary notation. Among these programing languages are: FORTRAN, COBAL, and FAP. FAP was used to program GASP. Here is a sample of FAP, with the English translation.

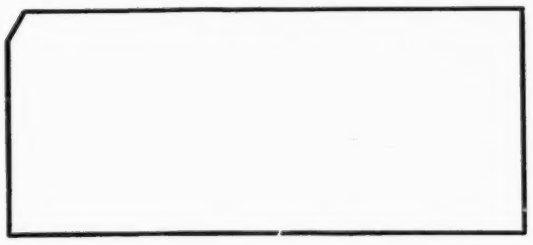
These three instructions direct the computer to see whether Class #1 and Class #2 conflict.

```
CAL CLASS 1
ANA CLASS 2
TZE NOCFLT

CONFLT  . . .        Conflict

NOCFLT  . . .        No Conflict
```

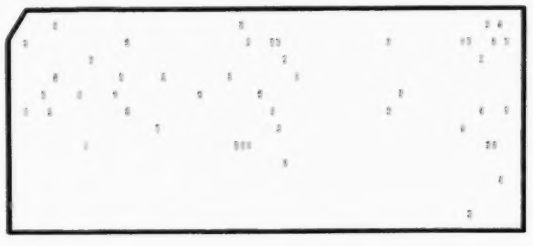
This sample of FAP looks complicated because it is unfamiliar. Certainly the 27 characters in the first three lines of the instructions above are easier to write and read than the 108 figures which would be needed to write the same fragment of program directly in binary.

Binary is that which characteristically has two values. Since a computer is an electronic device whose inner workings are coerced into action by a flow of electric current, it is capable of only two modes of expression. They are: yes, current is flowing in this direction, or, no, it is not.

Another example of binary is the presence or absence of holes, e.g., in an IBM card.



No holes = no



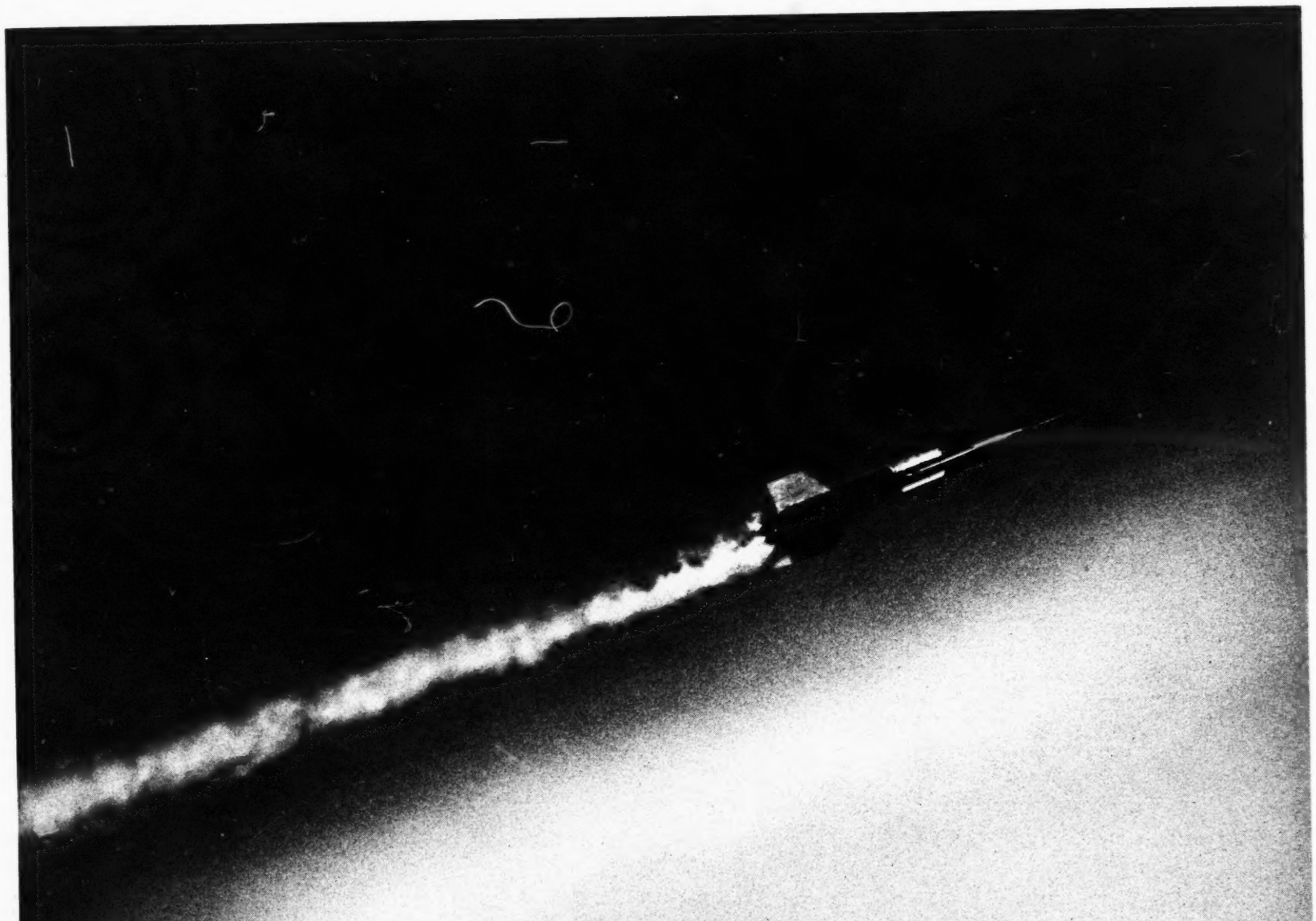
Holes = yes,

NO **EACH NO OR YES IS A BIT.** YES

A "word" in program language for the 7094 computer is coded into bits with a maximum of 36 allowed for each "word." GASP in binary looks like this:

G A S P **NO** LETTERS

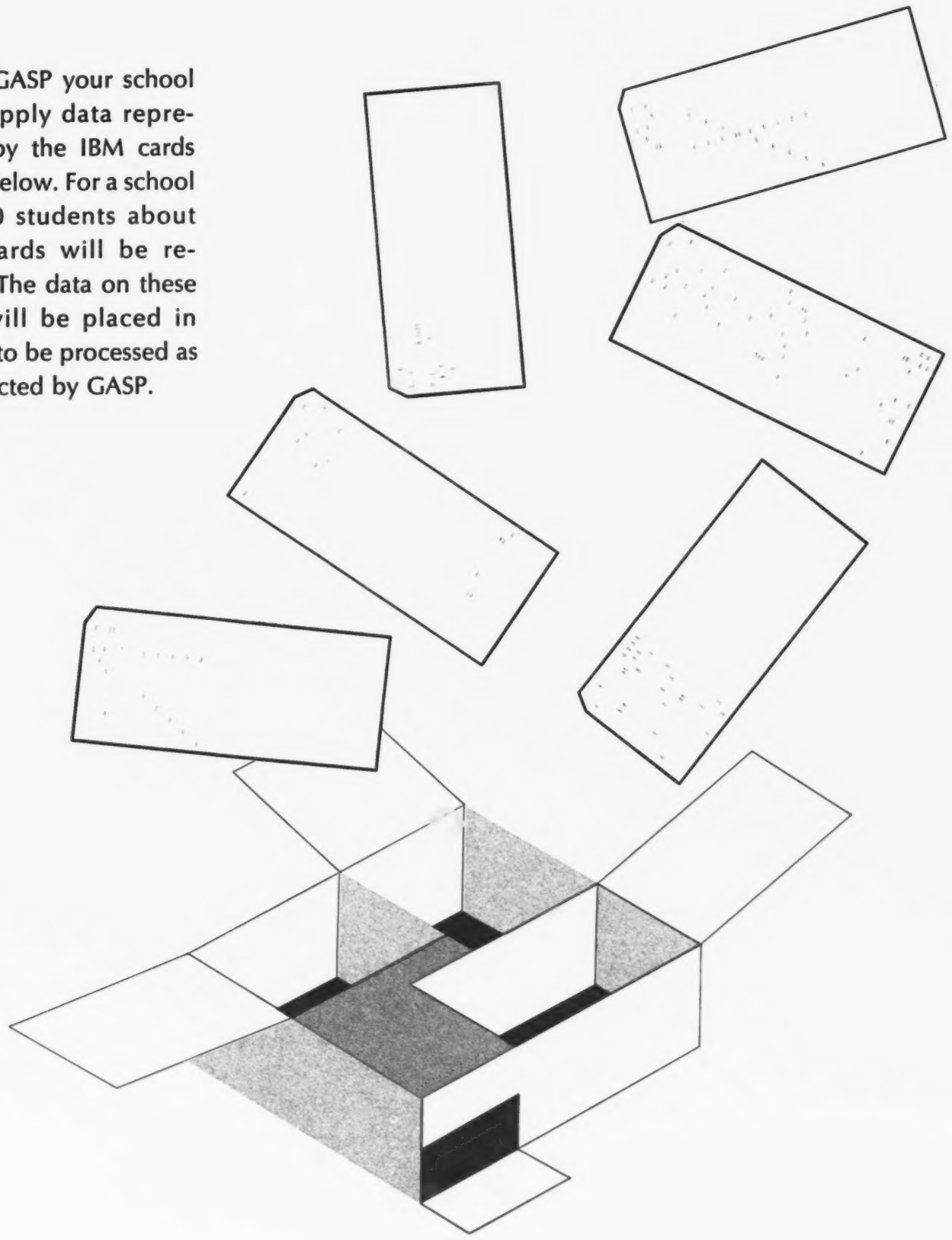
It is easy to see why the FAP vocabulary for GASP is limited to six-letter instructions which give rise to words such as NOCFLT (no conflict) in the example given above.



If it takes 24 bits just to record four letters, then complex information obviously requires an enormous quantity of bits. The 7094 computer used in the GASP program has a storage capacity of up to 32,768 words. Since electric impulses move at approximately the speed of light, it doesn't take long for a computer to search out and use indexed information. Items in storage can be located and made ready in a few millionths of a second. A millionth of a second is called a micro-second in computer parlance.

Thus: a rocket ship traveling 100,000 miles per hour moves 1¾ inches in a micro-second.

To use GASP your school must supply data represented by the IBM cards shown below. For a school of 1,000 students about 3,000 cards will be required. The data on these cards will be placed in storage to be processed as directed by GASP.



As an example, let's take Ted Lowry, eleventh-grade student, and see how the 7094 will store the information on his IBM cards concerning his assignments.

A There are holes in the cards which stand for Ted Lowry's name and subjects.

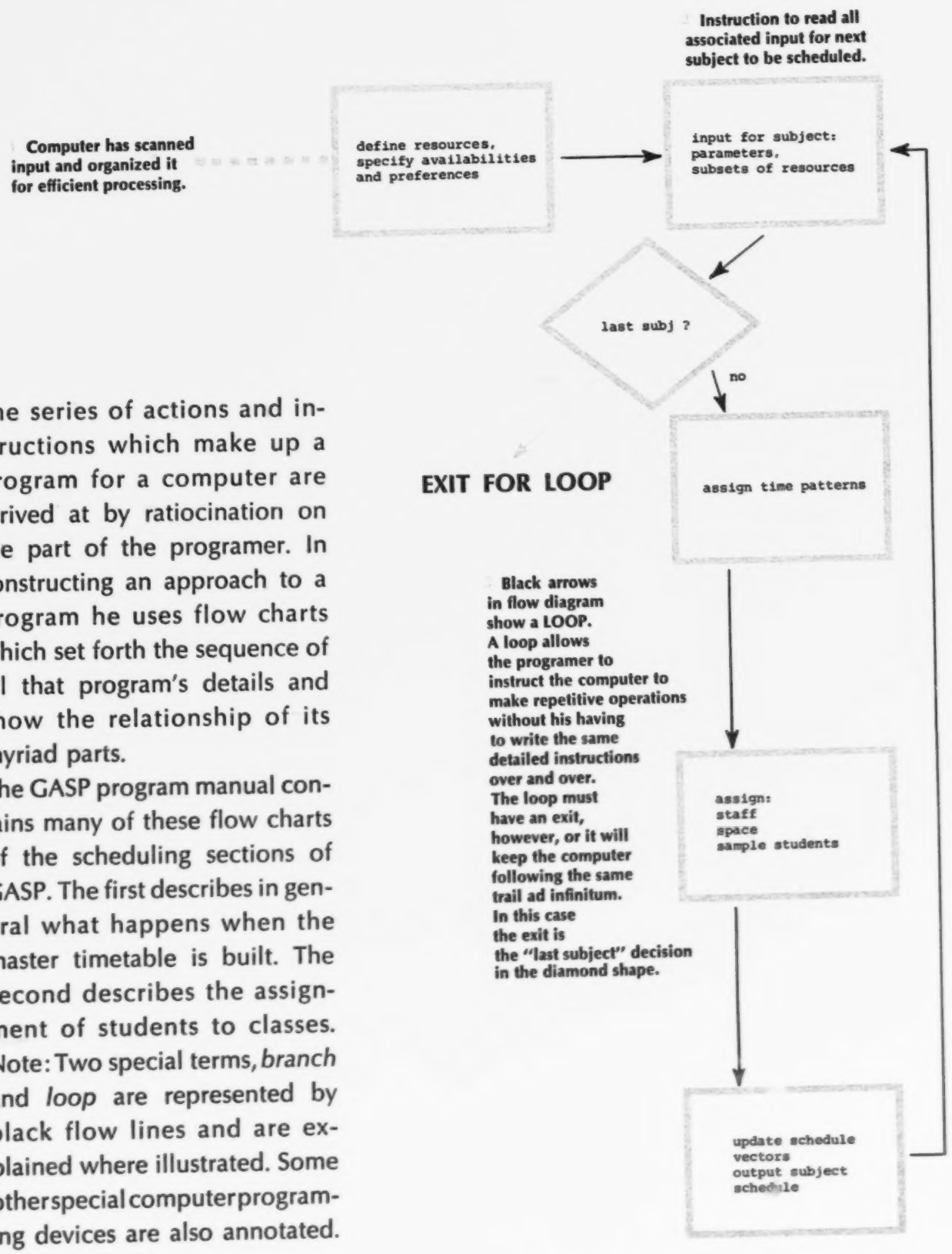
B If the school Lowry attends runs a seven-period day five days a week, then one word of 36 bits will suffice to represent his schedule (one bit is unused). Lowry has not yet been assigned to any classes, his schedule is blank to start with.

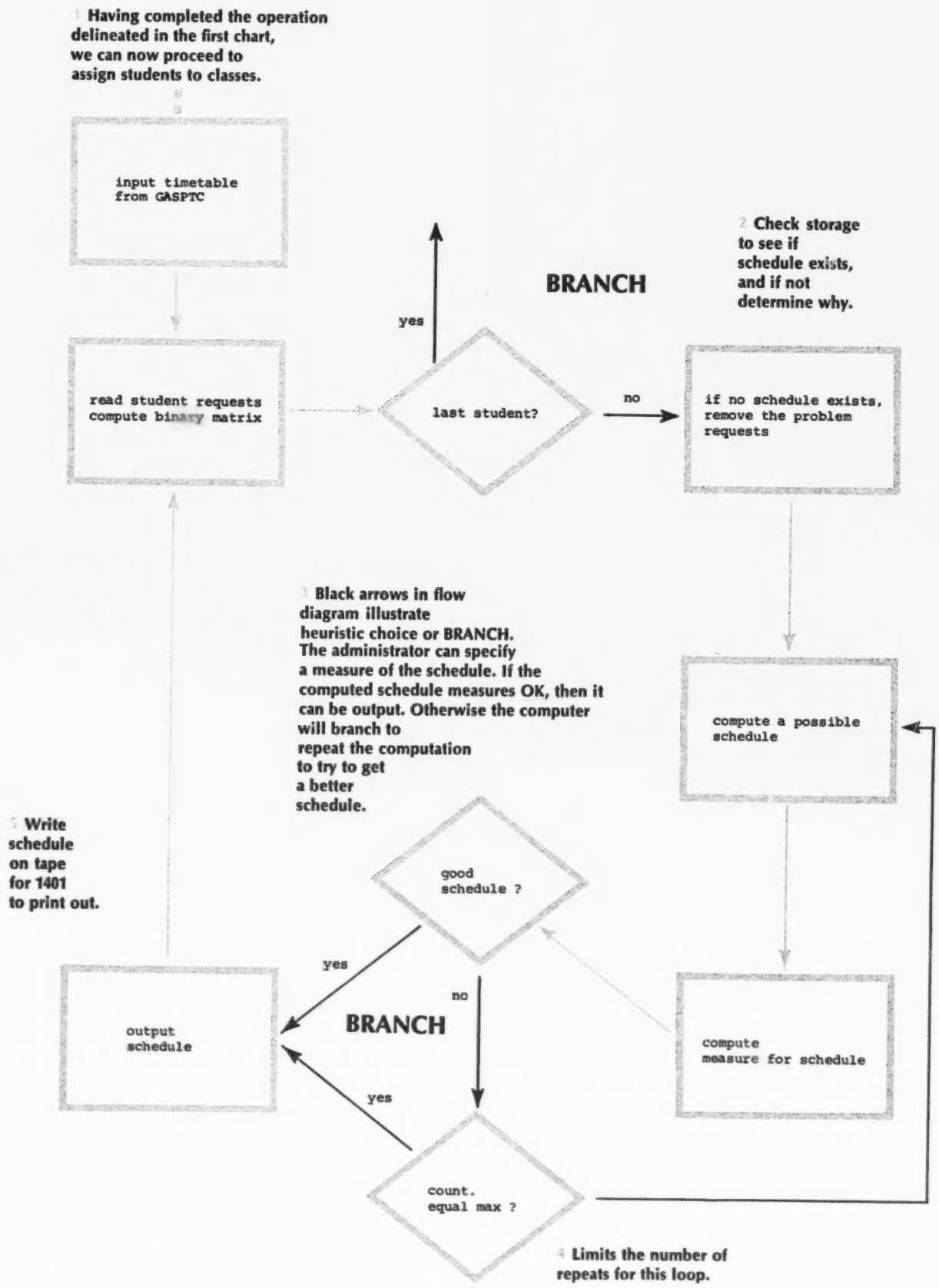
C Lowry has a job on Wednesday afternoons and must leave before the sixth period. We therefore read two bits into storage. The computer will now go along assigning Lowry to classes, study halls, lunch, etc. by putting in bits and checking to see what is left blank until all 35 positions are filled.

Notice the 0 in period 5 on Monday. Lowry has a free study period at a time when no study halls are available; this poses a minor conflict which requires human help to resolve.

The series of actions and instructions which make up a program for a computer are arrived at by ratiocination on the part of the programer. In constructing an approach to a program he uses flow charts which set forth the sequence of all that program's details and show the relationship of its myriad parts.

The GASP program manual contains many of these flow charts of the scheduling sections of GASP. The first describes in general what happens when the master timetable is built. The second describes the assignment of students to classes. Note: Two special terms, *branch* and *loop* are represented by black flow lines and are explained where illustrated. Some other special computer programming devices are also annotated.


Instruction to read all associated input for next subject to be scheduled.
Computer has scanned input and organized it for efficient processing.
define resources, specify availabilities and preferences
input for subject: parameters, subsets of resources
last subj ?
no
EXIT FOR LOOP
assign time patterns
Black arrows in flow diagram show a LOOP. A loop allows the programer to instruct the computer to make repetitive operations without his having to write the same detailed instructions over and over. The loop must have an exit, however, or it will keep the computer following the same trail ad infinitum. In this case the exit is the "last subject" decision in the diamond shape.
assign:
staff
space
sample students
update schedule
vectors
output subject
schedule


1 Having completed the operation delineated in the first chart, we can now proceed to assign students to classes.
input timetable from GASPTC
read student requests compute binary matrix
yes
BRANCH
last student?
no
2 Check storage to see if schedule exists, and if not determine why.
if no schedule exists, remove the problem requests
compute a possible schedule
3 Black arrows in flow diagram illustrate heuristic choice or BRANCH. The administrator can specify a measure of the schedule. If the computed schedule measures OK, then it can be output. Otherwise the computer will branch to repeat the computation to try to get a better schedule.
5 Write schedule on tape for 1401 to print out.
good schedule ?
yes
no
BRANCH
output schedule
yes
compute measure for schedule
count. equal max ?
4 Limits the number of repeats for this loop.

USE OF GASP FOR SCHEDULING

STEP 1 **Planning of schedule in accordance with school aims and policies**

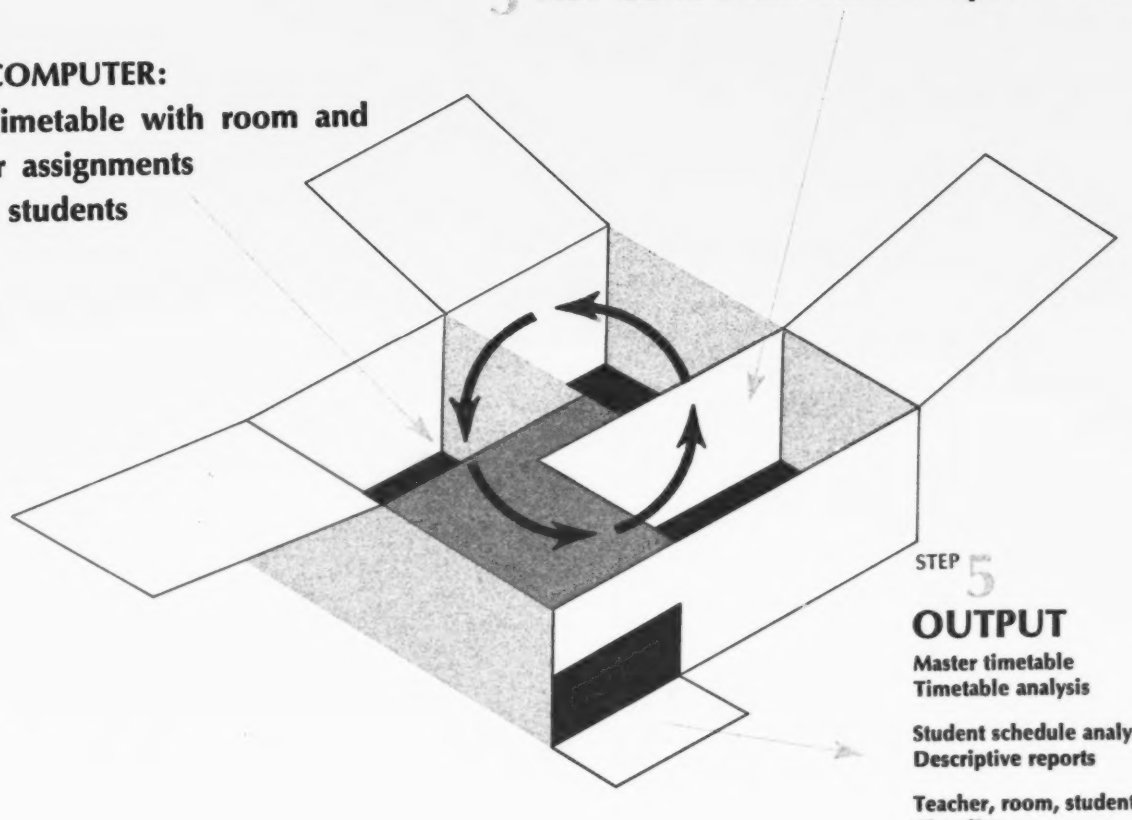
Staffing plan
Time allotments for each course
Student requests and groupings
Physical facilities

STEP 2 **Preparation of input**
Translation of plan into a form usable by the computer

STEP 3 **1401 COMPUTER: Process input for 7094**

STEP 4

7094 COMPUTER:
Build timetable with room and teacher assignments
Assign students



STEP 5

OUTPUT

Master timetable
Timetable analysis

Student schedule analysis
Descriptive reports

Teacher, room, student scheduling
Class lists

Analysis of computer run:

Decision made on whether or not to rerun (several man-hours of work) rerun (usually 10-20 runs)

GASP TIMETABLE ANALYSIS GASP TEST 1 PAGE 2 04 AUG

SUBJ	ND	NC	PW	RW	SW	PR	PS	PF	RR	RS	RF	SR	SS	SF
HI 2 S	2	3	.27	.32	.40	3	3	1.00	3	3	1.00	9	9	1.00
HI 2 S	2	3	.28	.32	.39	3	3	1.00	3	3	1.00	9	9	1.00
HI 3 S	6	9	.38	.44	.17	9	9	1.00	9	9	1.00	27	27	1.00
EL 3	2	2	.39	.10	.50	2	2	1.00	2	2	1.00	7	7	1.00
EL 3	2	2	.49	.10	.40	2	2	1.00	2	2	1.00	6	6	1.00
EL 1	2	2	.47	.16	.36	2	2	1.00	2	2	1.00	12	12	1.00
EL 2	2	2	.48	.16	.34	2	2	1.00	2	2	1.00	9	9	1.00
EL 3	2	2	.48	.15	.35	2	2	1.00	2	2	1.00	7	7	1.00
EL 3	3	3	.58	.08	.32	3	3	1.00	3	3	1.00	7	7	1.00
EL 2	3	3	.63	.06	.29	3	3	1.00	3	3	1.00	9	9	1.00
EL 2	2	3	.45	.10	.43	3	3	1.00	3	3	1.00	9	9	1.00
EL 1	4	4	.70	.07	.22	4	4	1.00	4	4	1.00	12	12	1.00
EL 1	2	4	.52	.11	.35	4	4	1.00	4	4	1.00	12	12	1.00
TOTALS														
70	159	203	.27	.38	.33	235	232	.98	203	203	1.00	1170	1170	1.00

STORAGE UNUSED BY THIS RUN (DECIMAL) EQUALS 13507 --GASA

ND = number of divisions scheduled

NC = number of sections or classes scheduled

PW,RW,SW = GASP parameters for subject indicating relative difficulty in scheduling instructors, rooms, and students to subject

PR = number of instructors required for classes of subject

PS = instructors actually scheduled for subject

PF = PS/PR

similarly for rooms (RR,RS,RF) and students (SR,SS,SF)

For each instructor, classroom, and student you specify any periods to be kept free from class assignments, or that you prefer to keep free; and you take the option of indicating maximum loads per week for students and instructors. You find that GASP allows considerable leeway in setting "parameters" for the scheduling of subjects. Thus you can, if you like, list just one allowable time pattern for a certain class (for example, "band or chorus, last period each day"), or you can let the program choose the best from a number of possibilities. Since your school's enrollment does not exceed 1,000, it would have been possible to include actual student requests in GASP's input; larger schools use, instead, a sampling of student requests. But since you want to build your schedule early, before actual student registrations are in, you also use this sampling technique.

Once this requisite information is assembled, the next job is to put it in shape for the computer. This means punching all the data into cards. Your school system, like many sizable systems in the U.S., has its own keypunch equipment, and has been handling many more or less standard data-processing jobs itself (such as printing grade reports and attendance rosters).

GASP STUDENT ANALYSIS RIDGEWOOD 12 PAGE 21 28 AUG 1964 JOB GASP

NAME	IX	NSUB	NDIV	SCHN	NSCH	MEAS	LOAD	SUBJECTS IN CONFLICT
LODDING ROBERT	81370	137	16	70	4	4	.00	85
MANN NANCY	81430	143	19	83	1	15	1.00	85
MARTIN DENNIS	81470	147	15	67	2	2	.00	75
MARTINO JOSEPH	81490	149	19	85	5	5	.00	87
MAYER KERRY	81530	153	17	66	2	2	.00	71
MAZUR ELIZABETH	81540	154	15	79	4	4	.00	71
MCKENNA TERRENCE	81560	156	17	71	5	15	1.00	75
MEISSNER PATRICIA	81570	157	16	70	2	2	.00	81
METTE JOHN	81580	158	16	72	1	15	2.00	75
NIEDERKORN BARBARA	81680	168	17	72	5	15	2.00	75
NOBLE MARY	81700	170	17	68	2	15	3.00	77
OSHEA DENNIS	81770	177	16	72	11	15	2.00	77
OSTLUND LINDA	81780	178	17	83	1	2	1.00	79
PALUMBO ANNETTE	81810	181	19	72	2	15	2.00	75
PAWINSKI JUDY	81830	183	17	67	1	1	.00	71
PERRY GREGORY	81840	184	15	67	2	2	.00	71
PIASECKI KENNETH	81860	186	15	86	1	15	3.00	75 55S
PINKA JUDY	81870	187	19	92	1	2	5.00	79 11S 55S
PISTONE ROSEMARY	81890	189	22	70	1	1	.00	77
PLACZEK GREGORY	81910	191	16	71	2	15	2.00	75
PRUSKI SUSAN	81980	198	16	70	1	1	.00	79
REICHEL DENNIS	82070	207	16	70	2	15	2.00	79
REIMANN MICHAEL	82080	208	16	72	14	15	2.00	75
RISKO ANDREA	82100	210	17					

Index

Specific measure for specific schools—lunches missed or passage time or preferred free time

Conflicts, these subjects were not scheduled

With all the school data punched on IBM cards and arranged in proper sequence, the GASP process is ready to begin. Your data requires something on the order of 3,000 IBM cards, making a stack 20 inches high. All this specific data is one segment of the necessary input. The other input is the program itself, GASP, which your consultant has obtained from Mr. Holz and his associates in the shape of a reel of magnetic tape ready to be read by the 7094. The program contains the *generalized* instructions that tell the machine precisely what to do with a school's specific data to produce a master schedule. There are upwards of 50,000 individual instructions in a program like GASP.

In effect, the *program* transforms the digital computer from a general purpose machine into a special purpose machine—in this instance, into a schedule-building machine. And it is because of the huge quantity of information (both data and instructions) that goes into the building of a school schedule that a computer with a large memory capacity, like the 7000 series, is required (GASP can be used interchangeably on the IBM 709, 7090, or 7094). This is the kind of computer often used in elaborate scientific work, or for complicated business calculations such as aircraft scheduling or inventory analysis. These large computers can perform calculations at fantastic speeds on a vast amount of information, all of it expressible in binary (i.e., "yes-no," or "on-off") terms; they are, however, relatively slow on input and output, besides being very expensive. By contrast, the smaller computers (such as the 1401), which are less expensive and are commonly used in business or

SUMMARY OF RUN RIDGEWOOD 12 PAGE 23 28 AUG 1964 JOB GASP
UTILIZATION REPORTS

NT = TOTAL NUMBER OF ELEMENTS IN SET REQUESTED.
NS = NUMBER OF ELEMENTS IN SET WHICH WERE SCHEDULED TO AT LEAST ONE CLASS.
FR = NS/NT, THE FRACTION OF THE SET USED.
PA = NUMBER OF PERIODS FOR WHICH ELEMENTS ARE SCHEDULED.
PT = NUMBER OF PERIODS WHICH WERE AVAILABLE FOR THE SET.
PS = NUMBER OF PERIODS WHICH WERE AVAILABLE FOR THAT PART OF SET WHICH HAD AT LEAST ONE ASSIGNMENT.
UT = PS/PA, UTILIZATION FOR COMPLETE SET OR SUBSET REQUESTED.
US = PS/PA, UTILIZATION FOR THAT PART OF SET WHICH WAS SCHEDULED TO AT LEAST ONE CLASS.

UTIL	STUD		US =	.78	UT =	.78	NS =	1138	NT =	1138	FR =	1.00	PS = 88985	PA =113800	PT =113800
UTIL	STUD	1A	US =	.86	UT =	.86	NS =	28	NT =	28	FR =	1.00	PS = 2425	PA = 2800	PT = 2800
UTIL	STUD	1B	US =	.82	UT =	.82	NS =	231	NT =	231	FR =	1.00	PS = 19090	PA = 23100	PT = 23100
UTIL	STUD	1C	US =	.87	UT =	.87	NS =	28	NT =	28	FR =	1.00	PS = 2440	PA = 2800	PT = 2800
UTIL	STUD	2A	US =	.80	UT =	.80	NS =	38	NT =	38	FR =	1.00	PS = 3067	PA = 3800	PT = 3800
UTIL	STUD	2B	US =	.80	UT =	.80	NS =	182	NT =	182	FR =	1.00	PS = 14587	PA = 18200	PT = 18200
UTIL	STUD	2C	US =	.80	UT =	.80	NS =	31	NT =	31	FR =	1.00	PS = 2484	PA = 3100	PT = 3100
UTIL	STUD	3A	US =	.76	UT =	.76	NS =	25	NT =	25	FR =	1.00	PS = 1903	PA = 2500	PT = 2500
UTIL	STUD	3B	US =	.77	UT =	.77	NS =	247	NT =	247	FR =	1.00	PS = 19046	PA = 24700	PT = 24700
UTIL	STUD	3C	US =	.76	UT =	.76	NS =	25	NT =	25	FR =	1.00	PS = 1919	PA = 2500	PT = 2500
UTIL	STUD	4A	US =	.69	UT =	.69	NS =	34	NT =	34	FR =	1.00	PS = 2361	PA = 3400	PT = 3400
UTIL	STUD	4B	US =	.72	UT =	.72	NS =	247	NT =	247	FR =	1.00	PS = 17996	PA = 24700	PT = 24700
UTIL	STUD	4C	US =	.76	UT =	.76	NS =	21	NT =	21	FR =	1.00	PS = 1602	PA = 2100	PT = 2100

First digit, class second, track level

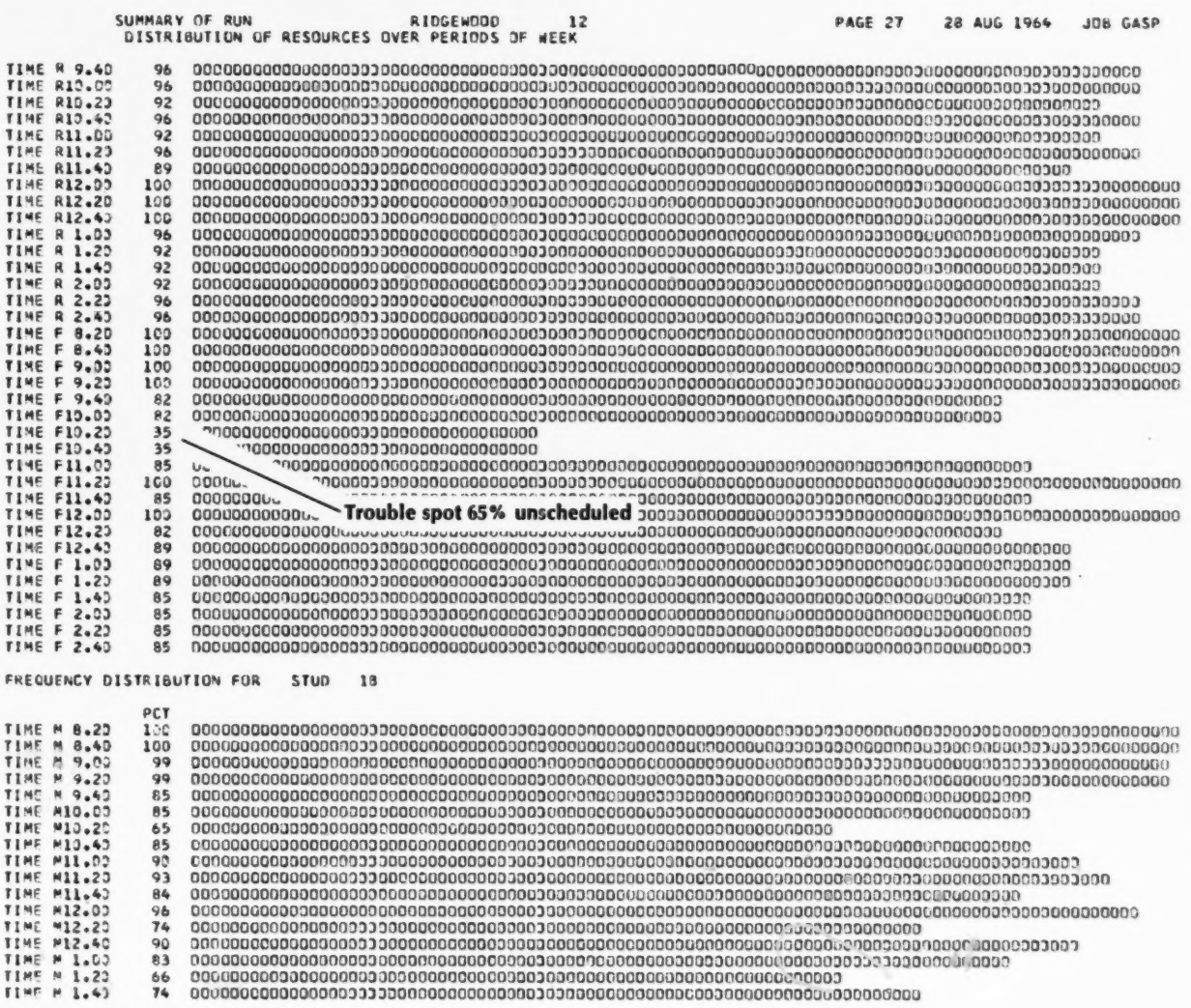
Percent time scheduled

Evaluation of schedule

Costs and measures (see key at top for explanation of terms) also available for classrooms and teachers

in routine institutional operations, can "read" decimal numbers directly and handle input and output very quickly; but with their small memories and control mechanisms, their use is confined to fairly simple processing of data like payrolls or bank accounts. Since many, if not most, operations require the special services of the smaller computers, all data-processing centers integrate large and small computers to provide a continuous process.

There is very little to see in the chaste, aseptic, air-cooled atmosphere of a large-scale data-processing center. What you see is the IBM cards fed deck by deck into the maw of the 1401. You see the tapes containing the various routines, subroutines, and data move and stop, move and stop, and small lights flashing softly on and off in an intricate pattern on the console. And you know, since you've been told so, that batteries of computer units, including the 1401, are handling the whole complex process continuously,



Histogram describing % people available in any grouping and period who have been scheduled

with little or no intervening human action: that, for example, the 1401 is converting the information about your school from the cards onto tapes for the 7094, concurrent with the inexorable flow of instructions and data through the machine's input, storage, and control units.

You also know (and, almost believe) that the 7094 memory has a capacity of 32,768 "words," and that each 36-bit word occupies one infinitesimal location in the memory core; you know that the machine takes two microseconds to perform a single action, and that the vast human jumble of your high school has even now been reduced to an infinity of 0's and 1's. But quicker than it takes to consider these enigmas, the first run is over: it has taken 18 minutes, and the 1401 is printing out the results with incredible speed. The output, now reconverted to decimal numbers, alphabetic text, and graphs, fills hundreds of large sheets of paper, and, with some expert help at the start, you find you can read the results quite easily.

TIMETABLE

SUBJ	TIME		ROOM
STCOUN	W	12.20	240
EN1B L	MR	8.20	148
EN1B S	TF	10.20	231
	TF	10.20	235
	TF	9.40	227
	TF	9.40	238
	MR	11.40	231
	MR	11.40	235
	TF	11.40	235
	TF	11.40	234
	MR	9.40	238
	MR	9.40	222
	TF	11.00	235
	TF	11.00	227
	MR	10.20	235

Time

Day

1 28 AUG 1964 GASD

INST	GRP	TIE	INDX	SA	SU
MISS TRACY	0	0	513	40	27
MRS BROWNELL	L 1	0	5	250	228
MRS BROWNELL	0	L 1	333	15	14
MISS MASLEY	0	L 1	334	15	14
MRS BROWNELL	0	L 1	335	15	15
MISS MASLEY	0	L 1	336	15	15
MRS BROWNELL	0	L 1	337	15	13
MISS MASLEY	0	L 1	338	15	13
MRS BROWNELL	0	L 1	339	15	15
MISS MASLEY	0	L 1	340	15	15
MRS BROWNELL	0	L 1	341	15	15
MISS MASLEY	0	L 1	342	15	15
MRS BROWNELL	0	L 1	343	15	11
MISS MASLEY	0	L 1	344	15	12
MRS BROWNELL	0	L 1	345	15	11

Seats used

Seats available

Basic schedule shows actual class sizes

On this first run, the computer has constructed a master timetable. It has formed it subject by subject, assigning instructors, space, students, and time periods. Originally, schools using GASP "told" the program the order in which to schedule subjects. Today, however (an example of the improvements experience has contributed to the revised GASP now in use), the machine is programed to make its own judgment of the best sequence. As one of its first tasks, the computer scans all relevant data and picks out as the first subject to schedule the one that looks the toughest, and so on down the line. For example, the machine is apt to schedule first all courses meeting in a single section. The scheduler may, however, control the sequence as much as he likes.

Because the program tabulates instructors, classrooms, and students continuously as it schedules subjects, it keeps a running check on conflicts and on periods assigned. As you were warned to expect, the results of the first run are somewhat discouraging. The first GASP run, and especially for a first-time user, is largely a trial run to show up correctable mistakes—clerical mistakes, for instance, that are almost unavoidable in punching 3,000 cards, or more important mistakes stemming from inadequate information or from inconsistent or inadequate planning by the scheduler (meaning you).

On this first run, you discover that clerical errors were made, and that a few dozen cards have to be re-punched. You may find, as well, a few conflicts that the computer was unable to resolve but which can be scheduled manually. Now you begin to understand what Holz means by the "rather intimate man-computer relationship" posited by the use of GASP. Studying

GASP FILE GASP 28 AUG 1964 RIDGEWOOD 12

Time: 11.00 Room: 212

SC1B M TF 11.00 212 MR WILLIAMS

Subject: SC1B Days: M TF Teacher: MR WILLIAMS

BALICE JULIA	80090
BURNS ROBERT	80200
COREY DANIEL	80380
DAVI JOSEPHINE	80440
GRISWOLD GLORIA	80860
KOBYLEWSKI CHRISTINE	81150
KSANDR GAIL	81240
MCDEROW JOYCE	81620
PALUMBO ANNETTE	81810
SEAMANN PAMELA	82240
STODULA MAUREEN	82380
TUHOLSKI MARIE	82520
VOSERGAU JOHN	82610
WARREN BARBARA	82630
WENDOLOWSKI DUANE	82670
RITTER CAROL	82105

Total class size

LS COUNT 16

the first returns, you find that in some cases where the machine gave up in making a required assignment, it is because your coded instructions have made the choice of classrooms, say, for a given subject so narrow that conflict is inevitable. It appears, also, that the "ties" that your data required between large, medium, and small social science classes make impossible demands on the classrooms available on Thursdays.

In sum, the first run indicates the need for certain clerical revisions, plus further administrative thought. Even on this first run, though, a fair number of subjects are up to 90 per cent assigned, or better. Others—especially where errors threw the machine off—are as low as zero. You and your assistants now devote considerable time, over a period of days, to studying the embryo timetable, getting the input data corrected, and adjusting certain of the conflicts "by hand."

So far, the schools using GASP have found that they can achieve a good master schedule, complete with student assignments, in anywhere from 7 to 20 runs of the program. Holz and his associates recommend spacing these runs a day or two apart, to give the scheduler time to analyze the reports from the previous run and to change whatever he has to for the next run—some changes reflecting compromises in the school's original intentions and some correcting improperly prepared material.

The second run, while it produces much better results, is still a trifle

discouraging; there are still some subjects to which instructors, rooms, or students are unassigned, and a few more data errors have turned up. But you have a schedule which is closer to one that will work, and, with four more runs, what appears to be a really good timetable. In between runs, as after the first, you and your staff work over the reports; in some cases you give changes to the program, in others you simply make adjustments yourself. After the last run, you add a few final touches. Then, later on when all student requests are in, GASP registers the students into the classes of the timetable. Total computer time: 3 hours and 20 minutes on the 7094, not quite 30 hours on the 1401. Total school staff time: about 6 man-weeks—or a quarter of the time required to produce a manual schedule. Your output includes the master timetable itself, class lists, individual teacher, room, and student schedules, and various descriptive materials that GASP can produce if desired, such as room utilization reports.

GASP handles students in such a way as to keep the size of sections balanced insofar as possible. In assigning students to classes, GASP scores close to 95 per cent success, depending on the complexity of the schedule and the amount of elective choice allowed students. The unassigned students you take care of manually by such devices as adding one more section in tenth-grade English, making certain acceptable shifts in the ability groupings for math, or disallowing a few elective requests. Like most school administrators, you have waited to assign students till the end of August in order to take account of late changes in student plans (the results, for instance, of going or not going to summer school), and these, as with the conventional schedule, you take care of by hand.

In building college schedules—as, for example, at MIT—GASP takes account of student preferences such as the time distribution of classes or choice of instructor, and other matters like passing distances between successive classes. In the process, the program may compute as many as 25 schedules for each student, all liable to be quite different from each other. The program then proceeds to pick the best of the schedules, in the light of the established criteria, and prints it out. This choosing among alternatives is an example of a computerized "heuristic," as against an "algorithmic," procedure. That is to say, the computer can be instructed to apply either algorithms (pro-

cedures, such as addition, that guarantee results) or heuristics (tentative procedures that may or may not produce satisfactory results). There is no reason that GASP could not schedule high schools, as well as colleges, to take account of these various considerations, but so far no high school has elected to do so.

On balance, you are well pleased with your first automated schedule. The final cost was well within your budget. It is clear that as you gain experience with GASP, you can undoubtedly conserve computer time, and your own. Using GASP is quite unlike using a computer for payrolls, or grade reports, or other standard data-processing jobs, where the program completely controls the machine and a single run produces the clear-cut output required. The computer, that faithful but unimaginative clerk, can do only what it's told to do. It will make choices if it is given all possible alternatives and the precise rules for choosing among them. But it cannot choose an alternative that the user hasn't made available. And since, in a process as complex as scheduling, it is impossible to cover every alternative, the computer will always leave a measure of important arbitrations to its boss.

There is another salient aspect of GASP which you can count on for important help in planning the new high school. Once you have all your data prepared and your schedule set, you can modulate the program to answer all kinds of "what if" questions. Thus, for example, you can find out the advantages or disadvantages of changing the medium grouping in social science, or of scheduling band or physical education outside the usual school day, or of using larger blocks of time for certain subjects. The program is set up to make certain types of systematic changes that you specify on continuous, successive runs. In the case of the new school, you plan to exploit this feature in various ways. For one thing, you'll be able to glean very specific information on the amount and kinds of space needed to match your educational design. You can do this by holding fixed all your data except the data on classrooms, and instructing the program to increase the number of available classrooms continuously from a small number to a large number on successive runs. The output will then show the minimum rooms required by a given program. But even more important, by studying the output from all the runs, you'll be able to see precisely what contribution additional space

above the minimum will make to desired changes and improvements in your educational program.

Simulation in St. Louis

GASP, the reader may recall, stands for Generalized Academic Simulation Programs. In Mr. Holz's words, GASP "attempts to simulate, or mimic, the clerical aspects of typical, idealized manual scheduling procedures." In short, it belongs to a genus of computer programs that simulate human procedures to achieve their purposes. "Simulation" is also a feature of GASP in the more specialized computer-linked sense of the word: viz., the program can be used to simulate a proposed, but nonexistent, school, in order to determine optimum size, design, utilization, and so on. GASP was used for this purpose in its pre-production test period, when Davies and Wolf, Freeman and Flansburgh, associated architects, made simulation runs to determine the number of classrooms for a proposed junior high school in Natick, Massachusetts. The results showed that the planners could eliminate two classrooms while at the same time maintaining a utilization rate of 85 per cent (it was, however, finally determined not to eliminate these two rooms because of population growth).

In 1963, GASP (with EFL financing) played an even larger role in plans for a new community college for the St. Louis–St. Louis County Junior College District. District officials, planning a campus to accommodate 4,500 full-time students, were intent on avoiding the under-utilization common to so many colleges. The president of the junior college district, Joseph Cosand, worked out a master plan for Meramec Junior College, complete with the number and kind of rooms necessary for the curriculum they had in mind. Then he and John Tirrell, formerly vice president of the district, asked the Automation Center of McDonnell Aircraft, in St. Louis, to take the master plan and try to prove the feasibility of 80 per cent utilization for classrooms and 65 per cent for laboratories—and if such unheard-of utilization were not feasible, to recommend changes that would permit it. The district, in short, proposed to run their new junior college as efficiently as the average secondary school, and almost twice as efficiently as the average small college.

The original notion had been to simulate the Meramec campus on one of McDonnell's computers, much as the Automation Center had simulated complex aircraft and other industrial production systems. After studying the problem, however, the Center concluded that this kind of elaborate "day-to-day" simulation was unnecessary, and that all they would need to answer the district's questions would be the master schedule for the new campus. Using GASP to program their 7094, the McDonnell people made about 20 simulation runs, with striking results. By the final run they proved that the high utilization desired for Meramec was entirely possible. In fact, through trying out all kinds of variations on the successive computer runs, the experiment indicated modifications whereby Meramec could meet all the requirements of the master plan with an over-all reduction of 22 rooms.

The simulation runs produced a mass of detailed information on space, time patterns, and instructor and student schedules.[6] In summary, the results showed an over-all classroom utilization of 85 per cent of the available time (and 89 per cent of the available seats per class), and an average laboratory utilization of 66 per cent (and an average student-station utilization of 91 per cent). In the process, the computer program was also used to answer a number of useful "what if" questions, such as: "What effect will the elimination of the noon hour (that is, scheduling no classes from 12 to 1) have on the master schedule?" and "What effect will the permitting of 20 per cent of the professors to teach no classes the first two hours of the day have?"

The application of the GASP study to Meramec Community College has enabled the staff to proceed more confidently in planning the highest possible room and seat utilization for the campus. In particular, the study has allayed the fears of the architects, who had felt that such high utilization was impossible, chiefly because it departed so sharply from traditional college space utilization pattern.

The Junior College District is building two additional campuses—the Florissant Valley Community College for 4,500 students, and the St. Louis City Campus for 7,000 students. Both of these campuses are now being planned in accordance with the findings from the GASP study, which had originally been applied only to the Meramec campus.

6. *The complete report,* Room Utilization at the Meramec Community College, *may be obtained from the McDonnell Automation Center, Division of McDonnell Aircraft, Box 516, St. Louis 66, Missouri.*

It now appears, according to Dr. Cosand, that the District may save upwards of $10 million in construction costs for the three campuses by applying to all of them the utilization rates for both room and seat space established in the Meramec study.

The Future of Automated Scheduling

To sum up the record thus far: in less than two years of actual production runs, GASP has demonstrated conclusively that it is a powerful aid in the building of master schedules. The program has produced schedules that the schools have found superior to handmade schedules of the past, and it has conserved valuable administrative time in the process. It is clear, too, that if GASP can produce good schedules for schools undertaking complex experimental programs like Ridgewood's and Wayland's, it can schedule traditional schools with ease. Production to date has been fairly costly, as is bound to be true of almost any evolving innovation, but there is clear evidence of the eventual economy of scheduling by computer.

Robert Holz and his associates have made continual improvements in GASP as they gained experience through working with actual schools. The second version of GASP was used in several schools for advance production of their 1964–65 schedules, and was made generally available in the spring of 1964. It includes, among other revisions, a number of supporting programs that simplify the school's preparation of data and that make the program's output much more readable—closer to the format of the old handmade schedule. The revision also provides the means to handle constraints like Wayland's large-medium-small group ties, and to sort certain data sequences automatically for better scheduling results.

Users of GASP now know what the program can do and what it cannot do. It cannot make basic administrative decisions, it cannot solve knotty problems by inventing solutions, it cannot cure flaws inherent in the school's educational program, its planning, or its plant. "One hundred per cent space utilization, complete freedom of choice of subjects for students, and the satisfaction of all time preferences for instructors, are examples of ideals which will not be realized," says Mr. Holz.

GASP's capabilities are equally clear. Properly handled by its human collaborators, the program can produce the master schedule (and related

materials) with great speed and accuracy. In doing so, it looks into far more alternatives than the human scheduler could encompass unaided, and, because it scans so many alternatives, GASP can produce a schedule that relies on fewer arbitrary dispositions of students, instructors, and space. In this sense, the computer-built schedule may take better account of human factors than the schedule built by hand. But, as stated earlier, GASP's ability to scan and choose alternatives depends on the user: he must give the program all possible alternatives and the precise rules for choosing among them.

Experience with the program thus far suggests certain ground rules:

1. *The basic, comprehensive rule (which, well applied, subsumes the others) is this: before attempting to use GASP to build a schedule, the school administration must strive for the greatest possible clarity and explicitness in what it is trying to do—educational goals, priorities among them, specifics on such details as time pattern, length of school day, ability and other groupings.*

If the user carries out the spirit of Rule **1**, he is well on the way to success with GASP. Use of the program has made clear that the chief cause of difficulties (excessive runs, unforeseen conflicts, etc.) is inadequate preparation on the part of the user and undue reliance on the machine to solve problems beyond its ken.

Other ground rules largely stem from Rule **1**:

2. *Experimenting schoolmen, keen to create a "Trump" school from scratch, would do well to remember that it is impractical to consider a new instructional program simply as it affects the single student. Schools must keep constantly in mind the problems entailed in combining the "ideal" plans for each of 500 or 800 or 1,000 students and matching them to available instructors and available space.*

3. *To get the most from computerized scheduling, school people must not go overboard for promising new assumptions (e.g., that youngsters are so eager to learn that they will all happily stay after school for music and/or independent study), nor should they cling stubbornly to old assumptions (e.g., every instructor needs his own classroom but not his own office).*

4. *The importance of setting priorities for the schedule can hardly be exaggerated. If a school gives first priority, say, to a policy of wide student electives, then it must be willing to sacrifice other, lower-priority policies—as, for instance, the arbitrary division of instruction in groups of set sizes, and rigid "ties" among the groups. Sometimes experimenting schools, in their zeal, prescribe novel arrangements that in practice constrain staff and students instead of freeing them.*

5. *Schools should devote thoughtful planning to the technical matter of time patterns. While GASP, if properly instructed, will choose among available time patterns for a given course, it is not equipped to design the over-all pattern for the day and week. The GASP-produced schedule profits from the care a school takes in arriving at this over-all pattern, and in adhering to it through the computer runs. The traditional school, with seven nonoverlapping periods five days a week, is easy to schedule: the chance of conflict is one in seven. By contrast, a highly experimental school with large, medium, and small groups linked together, can raise the chance of conflict as high as one in two. For a good schedule, with few conflicts, it is important to keep time patterns from overlapping as much as possible; Ridgewood has shown that even a highly unconventional school with a modular schedule can do so.*

6. *To exploit the advantages of GASP, schools should parlay the time and money they have put into computerizing the master schedule into: **a**) feasibility studies of contemplated changes in the school program; and **b**) studies to guide the design and size of contemplated additions or new plants.*

As pointed out earlier, GASP in its second year of production was already in its first revision. Revisions will continue to be made, partly as the result of practical experience with GASP, partly as the result of advances in relevant theory. The changes to date have come about entirely from production experience with GASP. The basic algorithms and heuristics guiding the program have proved generally satisfactory. A further note is of interest on the principles underlying GASP, or any such simulation program. In Mr. Holz's words: "Simulation is open to criticism on the ground that it may simulate a system of procedures which is far from ideal; intellectually it seems more

desirable to develop algorithms based on a sound mathematical model, so that exact or optimal solutions to a problem would result.

"A number of exact techniques do exist which theoretically are applicable to school scheduling problems; however, the large size of any usual school schedule makes the use of these techniques impossible. Lacking the ideal solution, the pragmatic approach was chosen for GASP."

To a mathematician, the pragmatic approach is apt to be distasteful: it lacks elegance and sophistication. Certain other experiments in automated scheduling have clung more closely to theory than has GASP (see Appendix A). Mr. Holz and his associates do not propose to rest with their pragmatic solution. They anticipate improvements that will take off from statistical theory, "along with explorations now being made in such areas as linear programing, graph theory, binary matrix manipulations, and others." The application of theoretical advances to GASP should produce, among other things, better ways to measure and compare alternate schedules, and more precise guidance for the scheduler in planning "constraints" for the new schedule. Meantime, initial explorations of graph theory at MIT suggest that the "rough, quick techniques such as are used in GASP and other similar programs" produce results surprisingly nearer the best than might be expected.

More than a century and a quarter ago, an eccentric Englishman named Charles Babbage perfected his Analytical Engine, which embodied almost all the essential elements of the modern digital computer. Babbage never succeeded in getting his Engine produced, and he exhausted his fortune and his patience in the attempt. Nor did many of his contemporaries understand what he was up to. One of the few was Ada Augusta, Countess of Lovelace and daughter of the poet Byron. She not only understood the Engine, she could explain it lucidly and write programs for it. The misunderstandings that the Analytical Engine met inspired this heavily underlined warning from Lady Lovelace: "In considering any new subject, there is frequently a tendency, first to *overrate* what we find to be already interesting or remarkable; and secondly, by a sort of natural reaction, to *undervalue*

the true state of the case, when we do discover that our notions have surpassed those that were really tenable.

"The Analytical Engine has no pretensions whatever to *originate* anything. It can do whatever we *know how to order it* to perform. It can *follow* analysis; but it has no power of *anticipating* any analytical relations or truths. Its province is to assist us in making *available* what we are already acquainted with."

One can hardly imagine a more precise description of the misunderstandings inspired today by the electronic digital computer and its works, and it may serve as an eloquent *envoi* to readers of this report. There is no use *undervaluing* GASP, as ultraconservative schoolmen tend to do ("It can't be done"), nor in *overrating* it, as enthusiastic school reformers sometimes do ("Computers will solve all our problems"). The impossible will still remain impossible, but GASP will most effectively do "whatever we know how to order it to perform."

STUDENT SCHEDULE

S.A.O. | STUDENT NUMBER | LOCKER NUMBER | 2194 HOME ROOM | PADOVANO TONY

HOME ROOM MEETS DAILY FROM 8:10 TO 8:17

PERIOD	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
8:20	ART2AM 144	ART2AM 144		ART2AM 144	ART2AM 144
8:40	ART2AM 144	ART2AM 144		ART2AM 144	ART2AM 144
9:00	ART2AM 144	ART2AM 144	HI4B S 232	ART2AM 144	ART2AM 144
9.20			HI4B S 232		
9:40	MA4B L 150	MA4B L 150	HI4B L 148	MA4B L 150	MA4B L 150
10:00	MA4B L 150	MA4B L 150	HI4B L 148	MA4B L 150	MA4B L 150
10:20	EN4B S 222			EN4B S 222	
10:40	EN4B S 222			EN4B S 222	
11:00	EN4B L 148	SC4B L 148	LUNCH CAF	EN4B L 148	SC4B L 148
11:20	EN4B L 148	SC4B L 148		EN4B L 148	SC4B L 148
11:40	BOOK M 134	BOOK M 134	BOOK M 134	BOOK M 134	BOOK M 134
12:00	BOOK M 134	BOOK M 134	BOOK M 134	BOOK M 134	BOOK M 134
12:20	BOOK L 150	MA4B M 233	PED3 L 148	EH4B M 206	MA4B M 233
12:40	BOOK L 150	MA4B M 233	PED3 L 148	EH4B M 206	MA4B M 233
1:00	LUNCH CAF	LUNCH CAF	EH4B M 206	LUNCH CAF	LUNCH CAF
1:20	PED34M GYM	PED34M GYM	EH4B M 206	PED34M GYM	PED34M GYM
1:40	PED34M GYM	PED34M GYM	SC4B M 229	PED34M GYM	PED34M GYM
2:00	PED34M GYM	PED34M GYM	SC4B M 229	PED34M GYM	PED34M GYM
2:20		SC4B S 222	EH4B M 206	EH4B M 206	SC4B S 222
2:40		SC4B S 222	EH4B M 206	EH4B M 206	SC4B S 222

RIDGEWOOD HIGH SCHOOL
SCHEDULE OF CLASSES
1964 - 1965

APPENDICES

Appendix A: Other Computerized Scheduling and Sectioning Programs

It is not surprising that the first applications of big computers to school scheduling concentrated on the student assignment, or *sectioning*, phase. Given a conventional instructional program that changes little from year to year, as it does at many schools and colleges, the phase of scheduling that calls for the most work—much of it clerical—is student assignment. So this was an obvious candidate for machine processing. Initial experiments were applied to college scheduling. One of the earliest was James Blakesley's at Purdue University starting in 1956, with the wide-ranging aim of devising a computer system "for registering, scheduling, and assessing fees for all students as a management device directed toward the improvement in a student's choice of courses and the overall utilization of resources."

Other colleges and universities, including the University of Massachusetts, the University of Rhode Island, and The Massachusetts Institute of Technology, developed similar computer programs. International Business Machines worked out a 7090 package based on an early Purdue model. Subsequently, Notre Dame, the University of Illinois, and other institutions experimented with programs written for the IBM 709 or 7090 or smaller computers like the 1620 that could section students for the largest universities. Robert Holz developed such a program for MIT. The Purdue program, meantime, has been refined to schedule over 18,000 students and assess their fees in less than six hours of computer time, and to produce an assortment of class lists, grade cards, enrollment reports, and other management data in the bargain for general University use. Next step in the project, which is now in the experimental stage and which Purdue hopes will go into operation in 1965, is the computerized construction of a schedule of classes.

IBM's 7070 or 7040 library program CLASS (Class Load and Student Scheduling) is now used by many secondary schools. Ernest Anderson developed a somewhat similar program at Harvard, coded for the IBM 7090, which is used by a number of school systems in the New England Education Data System (NEEDS) as well as other schools across the country. Generally speaking, all such programs assume a fixed schedule of classes with times, rooms, and instructors assigned—a schedule that has been built in the traditional way, by hand not by computer.

In a typical application, some 25 school systems in the Detroit area use IBM's CLASS.

The Ann Arbor School system is one. As the first step, Ann Arbor uses a small computer to tally student requests and to produce a "conflict matrix" (the school system now farms this procedure out, hopes someday to have an IBM 1401 on the premises). The tally is achieved by putting in the 1401 each student's name and his requested courses (expressed in three-digit numbers); it simply shows how many students want any pair of courses. The conflict matrix (or graph, or grid) is built by feeding the 1401 selected courses, usually those having only one or two sections—in short, the courses likeliest to produce conflicts and hardest to schedule.

With this computer-produced information as a base (a good timesaving in itself), the Ann Arbor principal and his assistants proceed to build the master schedule by hand, in the traditional way, assigning teachers, rooms, time-periods. The school system's data-processing department keypunches all this information, plus student requests, for the computer. The input for the big computer is then ready: (1) the master schedule, which the computer reads and stores, and (2) each student's requested courses, which the computer reads and applies to the master schedule. Ann Arbor's first use of CLASS, in 1964, produced excellent results. The computer scheduled all but 73 students out of an enrollment of more than 2,500. Total 7040 time was 33 minutes.

CLASS produces a variety of outputs, some of them optional. Basic outputs are: (1) student conflicts; (2) updated master schedule printed out—complete with number of seats assigned and left empty in each class; and (3) individual student schedules printed out. The program can handle up to 21 periods a day and up to 1,200 course sections; it automatically balances the size of sections, schedules homerooms and lunch periods, and takes account of such variables as sequential study halls or early dismissal of students who work after school. Programs of this type are particularly valuable to very large, more or less conventional schools that may have as many as 20 sections of a single course. Such schools seldom have much trouble with student conflicts, so that building a program is relatively easy. It is the assignment of students that entails vast clerical drudgery in order to keep classes balanced.

It might be pointed out, moreover, that sectioning programs can be valuable in helping the school administrator to improve his timetable (i.e., the time assignments for the master schedule). Since a big computer can assign all students for a secondary school in a short time for a few dollars, it is practical to run student requests through several times, as is rarely true with manual sectioning. In the process, the scheduler can reduce conflicts between runs by manual adjustments to his time assignments. NEEDS scheduling, following Anderson's techniques, uses this approach to put finishing touches on the handmade master timetable, as do most computerized student-assignment programs. Through "iteration" (the repeated runs that computerized sectioning makes feasible), it is possible to improve the master schedule without attempting to program the schedule itself.

While these sectioning programs were being developed, other researchers were pursuing the more difficult task of automating the master schedule. Much of this work, such as the studies directed by Dr. Albert G. Holzman at the University of Pittsburgh, concentrated on building a theoretical model. The research group at the University of Pittsburgh has also conducted investigations on a linear-programing approach

and a heuristic approach to the generation of master schedules

A good deal of the research has been posited on the use of computers in flexible schools of the future where each student proceeds at his own pace, without grade limitations and with an individual (and constantly changing) schedule. Comprehensive investigations of computer-technology in education, including scheduling, are under way at various centers, including System Development Corporation in Santa Monica, California, and the California State Department of Education's new Center for Research and Development in Educational Data Processing at Sacramento.

One practical outcome of theoretical investigations of this kind is a scheduling program devised by Professor Robert Oakford and others at Stanford University's School of Education, under a grant from the Fund for the Advancement of Education. Like GASP, the Stanford School Scheduling System (known to its creators as "the system" or S^4) is a generalized program for constructing master schedules and assigning students. The Stanford researchers explored the logic of schedule generation and also experimented with pragmatic techniques. Ultimately, whatever the algorithmic differences, the two programs evolved as similar in major practical respects. There are some differences between the two programs, to be sure. GASP is equipped to schedule as many as 4,000 courses, compared to Stanford's 500, and an unlimited number of students as compared to Stanford's 3,000. On the other hand, the Stanford system can schedule up to 18 teachers per teaching team, as against GASP's 5 (useful for scheduling departmental meetings, extra-curricular activities, etc.), and Stanford, unlike GASP, can specify two alternatives for each course for each student. But from the practical point of view of the schoolman, the two programs are pretty much alike in performance, and differ little in their ability to handle the range of variables an experimenting school might want—with respect to number of class periods, variation in period length, maximum number of modules, daily or weekly meetings, ties among small, medium, and large groups, sequencing of group or course phases, etc.

In effect, the Stanford system eventually evolved as a pragmatic solution like that of GASP. In turn, GASP was able to improve its performance thanks to some of Oakford's theoretical work (the GASP manual, for instance, credits Oakford with "a more sophisticated scheme" for distributing students between divisions).

The Stanford system went into actual school operation about the same time as GASP—the school year 1963–64. To date, about twenty-five schools have been scheduled, in whole or part, by Stanford, five of them for the second year: three California schools, one in Oregon, and one in Nevada. One of the California schools, a high school in suburban San Jose, used the system only for student assignment. In Stockton, however, Lincoln High School built a master schedule for a new, experimental, flexible program with the help of the Stanford system. Homestead High School, in Sunnyvale, used it to construct the schedule for a program which, while somewhat more conventional than Lincoln's, is gradually incorporating many innovations, including team teaching.

The system's biggest job to date has been the scheduling of Marshall High School, in Portland, Oregon, which enrolls upwards of 2,000 students in a highly experimental program housed in a conventional building

in the central city. At the other extreme is the Virgin Valley High School, in Clark County, Nevada—a rural six-year secondary school with 150 students.

Marshall High fully challenged the Stanford system. In the words of the school administration: "It would be impossible to prepare schedules manually for all students in a program such as the one now in use in Marshall High School"—a program that involves a school day of 21 modules of 20 minutes each, independent study averaging one-third of each student's time, instruction divided among large, medium, and small groups, and other innovations.

The Stanford people are candid about the difficulties revealed by early runs—notably in sequencing classes in the same subject and in balancing class size. In general, problems in the first area resulted from the schools' failure to specify compatible time patterns and parameters; problems in the second area stemmed from Stanford's assumption that any serious imbalances could best be handled by administrative intervention. With these exceptions, Stanford reports that all the schools involved in the first year found general satisfaction in their automated schedules and are continuing in the experiment. Revisions in the system and further school experience in using it are expected to produce much better results.

Professors Robert N. Bush and Dwight W. Allen of Stanford's School of Education, co-directors of the scheduling experiment, emphasize the importance of computerized scheduling as a means of freeing the school program from conventional restraints and opening the way to flexibility and innovation. As Dr. Allen says: "Communication and transportation have made the vast power of the modern computer available to every school in America. The question is: do schoolmen have the courage and foresight to use the computer now that it is available? With many more students, with much more to teach, with few qualified teachers, the need for educational innovation has never been more acute. The power of the modern computer stands ready to assist." In sum, the Stanford people are deeply concerned that the educational potential of electronic data processing should not further entrench orthodox school instruction and organization.

Appendix B: Costs and Computers

GASP is still new enough to preclude strict cost accounting. As suggested in the text, the saving in expensive professional time that results from computerized scheduling might properly be subtracted from GASP's out-of-budget cost. In general, to produce a schedule using GASP takes only about one quarter of the man-hours required to produce a schedule by hand. No school, however, has attempted more than a rough estimate of comparative costs.

The basic costs of GASP fall into two categories: 1) purchase of computer time; 2) salary or fee for expert help. The program itself—the actual tapes and related instructions—costs nothing (in keeping with the special freemasonry of computerdom).

GASP requires the use of a large IBM computer (a 709, 7090, or 7094), plus a smaller computer (1401) operated in tandem. In the present state of computer art, programs are not adaptable to all computers —they work only on the computers (or family of computers) for which they were designed (it would take up to several months to recode a program like GASP for use on another kind of large computer). Many sizable school systems now rent by the year 1401's (or comparable computers) to do their own data processing. Not even the largest school systems, however, are renting large computers like the 7090 or 7094 at the present time. There are only about 200 of these machines in the world, and they rent for about $500,000 a year or more (compared to $60,000 for a 1401).

A school wishing to embark on GASP should shop around for a good buy in computer services. A nearby university is the logical first choice; in most universities, however, computers run on two or even three shifts, and are booked solid for months ahead. Another possibility is any good big scientific laboratory (a thousand employees or more); some of these institutions charge only for large-computer time and throw in the 1401 time free (since they have relatively small need for the vast printed output which is 1401's specialty).

Outside of such nonprofit sources, the school's best bet is a commercial data-processing center or the nearest big plant of a research-oriented industry (such as aircraft, petro-chemicals, communications, defense industries in general). To get their money's worth out of the annual computer rent they pay, such companies need to get maximum use out of the machines. They are therefore eager to sell computer time. It comes high: about $200 an hour for the 709

and up to $700 for the 7090 or 7094 (but since the 7094 is eight times as fast as the 709, it's a better buy in the long run). Commercial rates on the 1401 range from $40 to $75 an hour.

In GASP's brief history, costs have varied widely—from $1 to $10 per student. Some of this cost is fixed, and does not depend on the size of the school. The variation is a function of many factors: the complexity of the school program in question, the accuracy and completeness of the data the school provides, the unit price of computer time, the extent to which the school requires expert assistance, and the amount of output that the school wants. For instance, GASP scheduled one school with 2,500 students at about $3 per student, and a school with 400 at about $7 per student. In both cases, the cost included *expertise* plus multiple outputs, including grade reports and attendance records, as well as all the standard and optional GASP outputs.

A small school (500 students or less) may expect to need about one hour or less of large-computer time. A big school may need between three and four hours. As the text explains, GASP gets its results through a number of successive runs on the computer. A single run for a small school takes no more than 10 minutes; for a big school, 20 minutes. Whereas in the nature of the program there will always be more than a single run, it is evident that the runs needed to produce a good schedule can be reduced (perhaps to four or five) by careful preparation of data, thoughtful collaboration by the scheduler, and sheer experience in using the program.

The 1401, which processes input for the large computer and thereafter prints out the results for the scheduler, takes 20 to 30 hours to do a complete GASP job for a school of 1,000 students or more; nearly all of this time is accounted for by the printout of schedules, lists, analyses, and assorted reports. The 1401 time for a small school averages about 5 hours.

Besides computer time, the other basic GASP cost is for expert help. Here again there are various ways to skin the cat. A school may already employ a computer man, or it may hire one when it undertakes GASP, probably part-time or in conjunction with other schools. Or the school may retain a consultant for a month or so to manage the technical details of GASP and work with the scheduler, who should plan to spend twice that much time himself. The school may be able to buy the GASP scheduling service as a package from the company or institution from which it buys computer time (the McDonnell Automation Center in St. Louis, and International Telephone & Telegraph in northern New Jersey, for instance, are both currently servicing GASP in this fashion).

Obviously, there is room for considerable variation in cost among these various expedients, especially when correlated with all the variables itemized above in connection with the cost of computer time. The range might be from the $500 that a small school would be likely to pay for a few days of expert consultation, to the $7,000 or $8,000 in salary that a big school pays to a staff computer man for the time devoted to GASP and other data-processing work.

A minor element in GASP expenses is the preparation of the school's input data. For schools that already use data-processing techniques to turn out grade reports, class lists, and so on, the extra cost of preparing data for GASP will amount to no more than a few hundred dollars.

One thing is quite clear: as schools gain

experience in the use of GASP, costs go down markedly. For one thing, they learn to perfect their preparation of data, thereby minimizing mistakes and reducing the number of computer runs required. The scheduler becomes more adept and imaginative in interaction with his electronic partner, and develops a sixth sense that enhances his understanding of what the computer can and cannot do. Furthermore, after no more than one time around with GASP, an alert scheduler—lacking even a mathematics, much less a computing, background—catches on fast to the ins and outs of automized scheduling, and to what at the outset seems impenetrable to anybody outside the magic inner circle of computerdom. As noted in the text, Ridgewood High School, in Norridge, Illinois, after only one year's experience with GASP, now can handle pretty much the whole process, with minimum outside help, up to the actual computer runs. And Ridgewood's "expert," an administrative assistant to the principal, was formerly a teacher of industrial arts, with no great backlog of mathematics, who learned about GASP "without too much trouble"—mainly by working with the experts and boning up a bit on digital computers. He spends no more than one fourth of his time on scheduling matters, including GASP.

A final word on costs: *feasibility* studies like those mentioned in the text, which use GASP to analyze proposed changes or future space needs, cost much less than the production of actual schedules. For one thing, most of the time-consuming details that have to be settled in a production schedule are of small importance in studies of feasibility. And such studies require little extra data preparation if they follow an actual production run. Most feasibility studies should cost no more than a few hundred dollars.

The following publications are available from the offices of EFL: 477 Madison Avenue, New York, New York 10022.

BRICKS AND MORTARBOARDS A guide for the decision makers in higher education: How the colleges and universities can provide enough space for the burgeoning enrollments of this decade; how that space can be made adaptable to the inevitable changes in the educational process in the decades ahead. (One copy available without charge. Additional copies $1.00.)

COLLEGE STUDENTS LIVE HERE A report on the what, why, and how of college housing; reviews the factors involved in planning, building, and financing student residences.

THE COST OF A SCHOOLHOUSE A review of the factors contributing to the cost and effectiveness of schoolhousing, including planning, building, and financing.

DESIGN FOR ETV—PLANNING FOR SCHOOLS WITH TELEVISION A report on facilities, present and future, needed to accommodate instructional television and other new educational programs. Prepared for EFL by Dave Chapman, Inc., Industrial Design.

RELOCATABLE SCHOOL FACILITIES A survey of portable, demountable, mobile, and divisible schoolhousing in use in the United States and a plan for the future.

THE SCHOOL LIBRARY A report on facilities for independent study, with standards for the size of collections, seating capacity, and the nature of materials to be incorporated.

TO BUILD OR NOT TO BUILD A study of the utilization of instructional space in small liberal arts colleges, with a do-it-yourself workbook for the individual use of the institutions that wish to survey their own utilization levels.

PROFILES OF SIGNIFICANT SCHOOLS

A series of reports which provide information on some of the latest developments in school planning and design.

Belaire Elementary School, San Angelo, Texas
Heathcote Elementary School, Scarsdale, New York
Montrose Elementary School, Laredo, Texas
Two Middle Schools, Saginaw Township, Michigan
Newton South High School, Newton, Massachusetts
Holland High School, Holland, Michigan

Schools for Team Teaching–ten representative examples
High Schools 1962—educational change and architectural consequence

CASE STUDIES OF EDUCATIONAL FACILITIES

A series of reports which provide information on specific solutions to problems in school planning, design, and construction.

1. CONVENTIONAL GYMNASIUM VS. GEODESIC FIELD HOUSE A comparison of cost, space, and advantages based on a case study of West Bethesda High School, Montgomery County, Maryland.

2. SPACE AND DOLLARS: AN URBAN UNIVERSITY EXPANDS A report on the economical physical expansion of urban universities based on a case study of Drexel Institute of Technology.

3. LABORATORIES AND CLASSROOMS FOR HIGH SCHOOL PHYSICS Chapter reprinted from MODERN PHYSICS BUILDINGS: DESIGN AND FUNCTION.

4. A DIVISIBLE AUDITORIUM/BOULDER CITY, NEVADA Case study of an auditorium that can be converted to instructional spaces by the use of soundproof, operable walls.

5. NEW CAMPUSES FOR OLD: A CASE STUDY OF FOUR COLLEGES THAT MOVED What the decision to move means from an economic, academic, social, and physical point of view.

6. A COLLEGE HEALTH CENTER Case study of a model center for small private colleges; architectural design by Caudill, Rowlett & Scott.

7. NEW BUILDING ON CAMPUS: SIX DESIGNS FOR A COLLEGE COMMUNICATIONS CENTER Graphic representations of the results of an architectural competition for a new space to house the accoutrements of instructional aids and media.

8. THE SCHOOLS AND URBAN RENEWAL A case study of the Wooster Square renewal project in New Haven, Connecticut.

9. AIR STRUCTURES FOR SCHOOL SPORTS A study of air-supported shelters as housing for playfields, swimming pools, and other physical education activities.

TECHNICAL REPORTS

1. ACOUSTICAL ENVIRONMENT OF SCHOOL BUILDINGS by John Lyon Reid and Dariel Fitzroy—Acoustics of academic space in schools. An analysis of the statistical data gathered from measurement and study.

COLLEGE NEWSLETTER A periodical on design questions for colleges and universities.

REPORT DESIGNED BY: *Sutter & Wartik* PRINTED BY: *Georgian Lithographers*

11020	626	12	24	1	15	2.00
80010	1	16	41	10	15	2.00
80020	2	14	36	1	1	.00
80050	5	16	46	2	2	.00
80080	8	14	36	1	1	.00
80320	32	16	47	1	1	.00
80400	40	16	45	2	2	.00
80520	52	13	34	1	1	.00
80550	55	15	41	1	1	.00
80580	58	16	55	2	2	.00
80640	64	16	53	2	2	.00
80660	66	12	28	1	1	.00
80730	73	15	51	1	1	.00
	76	12	29	1	1	.00
	77	14	40	1	1	.00
	80	11	40	3	3	.00
	84	15	35	1	1	.00
	88	14	35	1	1	.00
	89	14	39	1	1	.00
80960	96	12	31	1	1	.00
80970	97	16	52	3	3	.00
81060	106	16	42	1	15	2.00
81070	107	16	45	1	1	.00
81100	110	16	52	1	1	.00
81110	111	16	55	2	2	.00
81210	121	16	43	1	1	.00
81220	122	15	41	2	2	.00
80280	128	15	39	1	1	.00
81330	133	12	32	1	1	.00
81450	145	10	38	1	1	.00
81500	150	16	47	1	1	.00
81510	151	14	36	1	1	.00
81520	152	12	32	1	1	.00
81650	165	14	46	2	2	.00
81660	166	16	49	2	2	.00
81670	167	16	44	1	1	.00
81760	176	13	34	1	1	.00
81850	185	13	34	1	1	.00
81900	190	15	51	5	5	.00
81940	194	13	43	6	6	.00